DX Application Acceleration Platform

Installation and Administration Guide for DXOS Version 5.0

Revision 1.00

Juniper Networks, Inc.

1194 North Mathilda Avenue

Sunnyvale, CA 94089

USA

408-745-2000

www.juniper.net

Part Number: 530-014646-00, Revision 1.00

DX Application Acceleration Platform Installation and Administration Guide
Copyright © 2005, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Writing: Writers Works
Editing: Writers Works
Illustration: Writers Works
Cover Design: Edmonds Design

Revision History

10 October, 2005—Revision 1.00 First Official Release

The information in this document is current as of the date listed in the revision history.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

Products made or sold by Juniper Networks (including the ERX-310, ERX-705, ERX-710, ERX-1410, ERX-1440, M5, M7i, M10, M10i, M20, M40, M40e,M160, M320, and T320 routers, T640 routing node, and the JUNOS and SDX-300 software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479,6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year2038. However, the NTP application is known to have some difficulty in the year 2036.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING,INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMEROR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THESOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer anon-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

a. Customer shall use the Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller, unless the applicable Juniper documentation expressly permits installation on non-Juniper equipment.

b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees.

c. Other Juniper documentation for the Software (such as product purchase documents, documents accompanying the product, the Software user manual(s), Juniper's website for the Software, or messages displayed by the Software) may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, concurrent users, sessions, subscribers, nodes, or transactions, or require the purchase of separate licenses to use particular features, functionalities, or capabilities, or provide temporal or geographical limits.Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Software on non-Juniper equipment where the Juniper documentation does not expressly permit installation on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; or (k) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. If the Software is distributed on physical media (such as CD), Juniper warrants for 90 days from delivery that the media on which the Software is delivered will be free of defects in material and workmanship under normal use. This limited warranty extends only to the Customer. Except as may be expressly provided in separate documentation from Juniper, no other warranties apply to the Software, and the Software is otherwise provided AS IS. Customer assumes all risks arising from use of the Software. Customer's sole remedy and Juniper's entire liability under this limited warranty is that Juniper, at its option, will repair or replace the media containing the Software, or provide a refund, provided that Customer makes a proper warranty claim to Juniper, in writing, within the warranty period. Nothing in this Agreement shall give rise to any obligation to support the Software. Any such support shall be governed by a separate, written agreement. To the maximum extent permitted by law, Juniper shall not be liable for any liability for lost profits, loss of data or costs or procurement of substitute goods or services, or for any special, indirect, or consequential damages arising out of this Agreement, the Software, or any Juniper or Juniper-supplied software. In no event shall Juniper be liable for damages arising from unauthorized or improper use of any Juniper or Juniper-supplied software.

EXCEPT AS EXPRESSLY PROVIDED HEREIN OR IN SEPARATE DOCUMENTATION PROVIDED FROM JUNIPER AND TO THE EXTENT PERMITTED BYLAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE),INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR ORINTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the

license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to you may contain encryption or other capabilities restricting your ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4,FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement.

If you have any questions about this agreement, contact Juniper Networks at the following address:

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
Attn: Contracts Administrator

Table of Contents

List of Figures

List of Tables

About This Guide

This document provides an overview of how to install and configure the DX Application Acceleration Platform. This document applies to all DX Application Acceleration Platform product models. Topics discussed include:

Audience

This document assumes that the reader has knowledge of the network architecture or topology in which the DX appliance will be installed. This documentation is intended for network engineers, web operations engineers, IT professionals, and system administrators who have experience with the following:

- Installing, configuring, and administering network equipment
- Managing web traffic and connectivity

Conventions

Table 1 illustrates the conventions that are used in this manual.

Table 1: Notation Conventions

Notation	Example	Meaning and Use
`Courier typeface`	`.ini` file	Code listings, names of files, symbols, and directories, are shown in courier typeface.
`Bold Courier typeface`	install	In a command line, keywords are shown in bold, non-italic, Courier typeface. Enter them exactly as shown.
Square brackets	[*version*]	You may, but need not, select one item enclosed within brackets. Do not enter the brackets.
Angle brackets	<*username*>	You must provide the information enclosed within brackets. Do not enter the brackets.
Bar	les | les.out	You may select one (but not more than one) item from a list separated by bars. Do not enter the bar.

When listings are shown of computer output, an effort has been made not to break up the lines when at all possible. This is to improve the clarity of the printout. For this reason, some listings will be indented, and others will start at the left edge of the column.

Cluster, Redirector, Forwarder, Cache, and ActiveN Group Naming Conventions

This feature allows you to name a Cluster, Redirector, Forwarder ("cluster" in its general sense), cache, or ActiveN group to enhance the usability of the DX appliance. A default name will be assigned when a name is not provided. It will be most useful for medium to large customers that have multiple clusters and need easier identification (e.g., meaningful identifier instead of a number) for ease of management. In addition, this feature solves the problem of cluster renumbering when a cluster is deleted.

You can name a cluster, redirector, or forwarder at creation or after it is created. You can also rename an existing cluster, redirector, or forwarder. Names are subject to these restrictions:

- Names can be up to 32 characters long.
- The strings "all," "cache," and "NULL" are reserved names and must not be used as a cluster, cache, or ActiveN group names.
- Names are case-sensitive, except for the reserved names "all," "cache," and "NULL". No variations of these words can be used.
- Names can be any valid character string and may be integer-only. The valid characters are:
  - @;$^&*() = +!< > ,[]/_.-0123456789
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
- The name cannot contain white space.
- When a cluster, redirector, or forwarder is created without a name specified, a name is automatically created. The name for this unnamed cluster follows the previous behavior as much as possible. Configuration exports from previous releases contain the number of the cluster in the add command, and the remaining cluster configuration commands in the export depend upon the implied identifier of 1,2,3, . . . Using the next available integer as the implied name for a cluster mimics the behavior in previous releases. This way, imports of configurations from previous releases continue to function.
- You can not create a new cluster, redirector, or forwarder if the specified name is already in use. The name space that is considered for name collisions is limited to the type of cluster being added, e.g., when adding a forwarder, the DX appliance will only examine the names of other forwarders for collisions. This allows a cluster, redirector, and forwarder to all share a name of "1." This is needed for backwards compatibility.
- All references to cluster, redirector, and forwarder use a name instead of a numbered index. The ability to refer to a cluster by index will no longer be supported.

Some examples are:

```
% set cluster <N> …' becomes '% set cluster <name> …
% show cluster <N> …' becomes '% show cluster <name> …
% delete cluster <N> …' becomes '% delete cluster <name> …
% set redirector <N> …' becomes '% set redirector <name> …
% show redirector <N> …' becomes '% show redirector <name> …
% delete redirector <N> …' becomes '% delete redirector <name> …
% set forwarder <N> …' becomes '% set forwarder <name> …
% show forwarder <N> …' becomes '% show forwarder <name> …
% delete forwarder <N> …' becomes '% delete forwarder <name> …
```

- Integer-only names are assigned when no name is specified. The next available lowest integer is used for the assigned names. Example: if you add four clusters without names, the clusters “1”, “2”, “3”, and “4” will be created. If you then delete cluster “2,” the remaining clusters names will not change, leaving clusters “1”, “3”, and “4”. If you then add another cluster without a specified name, the assigned name will be “2” since this is the next lowest available integer. This is referred to as “filling the holes,” and is different from the previous behavior where after deleting cluster 2, the cluster numbers collapsed leaving clusters “1”, “2”, and “3”, and the new cluster's number would then be “4”. This is because all clusters are now referred to by name instead of index.

- The cluster name is included as part of the “`add`” command on a configuration export.

- The sort order for display of clusters (including tab completion) mimics “sort -n” behavior. This sorts the names according to arithmetic value for any and all leading numeric values in a name. Example: 23www will be listed before 3abc, and 9 will be listed before 11.

As an additional assistance for identification and purpose of clusters, redirectors, and forwarders, a “note” can be applied to individual clusters. This note is limited to 512 characters, and is expected to be free-form text but may not include new lines. This allows administrators to fully describe the cluster's usage, contact information, warnings, or any other pertinent information deemed necessary.

Optional Features

Certain features within the Juniper product line are optional. They are enabled through the use of a “License Key”. If you wish to enable any of these optional features, contact your Juniper Sales Representative.

Table 2: Optional Features

Feature
OverDrive
3G Cache

Chapter 1
Introduction

This chapter provides an introduction to the DX Application Acceleration Platform, discussing the following topics:

Overview

The DX Application Acceleration Platform represents a new concept in web server acceleration. It addresses the inefficiencies in server architecture, network architecture, and network protocols that limit the performance of your web site and web servers. The DX appliance solves these inefficiencies by providing its own highly-optimized network architecture and breakthrough data optimization and connection handling capabilities to make your web pages download faster and your web servers more efficient than ever before.

With the DX appliance installed between your web servers and router/firewall, your site pages will reach end-users 2X-4X faster, and your web servers will experience a tenfold increase in capacity combined with an increase in your bandwidth efficiency.

Package Contents

The DX appliance ships with the following items. If any of these items are missing or damaged, please contact a Juniper Networks Customer Service Representative to obtain a replacement.

- One DX Application Acelleration Platform
- One AC Power Cord
- One Ethernet Cable
- One Null-Modem Cable
- One Rack Mount Kit (rack ears and screws)
- One *DX Application Acceleration Platform Quick Start Guide*
- One CD-ROM containing the following manuals in Adobe Acrobat format:
  - *DX Application Acceleration Platform Quick Start Guide*
  - *DX Application Acceleration Platform Installation and Administration Guide*
  - *DX Application Acceleration Platform Command Line Reference* Guide

Installation Overview

Installation requires adding no hardware or software to your web servers. It also requires no modification or preparation of the content to be accelerated. Of course, the DX appliance is completely transparent to end users, requiring no special plug-in or software download.

This is a high-level overview of the steps required to install the DX appliance:

- Connect the power and network cables.
- Connect the DX appliance console port to a terminal or a computer with a terminal emulation program, then provide the DX appliance with basic network and target host information.
- Integrate the DX appliance into your web traffic flow.

DX Appliance Hardware

Figure 1 shows a front view of the DX appliance 1U Chassis.

Figure 1: Front View of the DX Appliance 1U Chassis

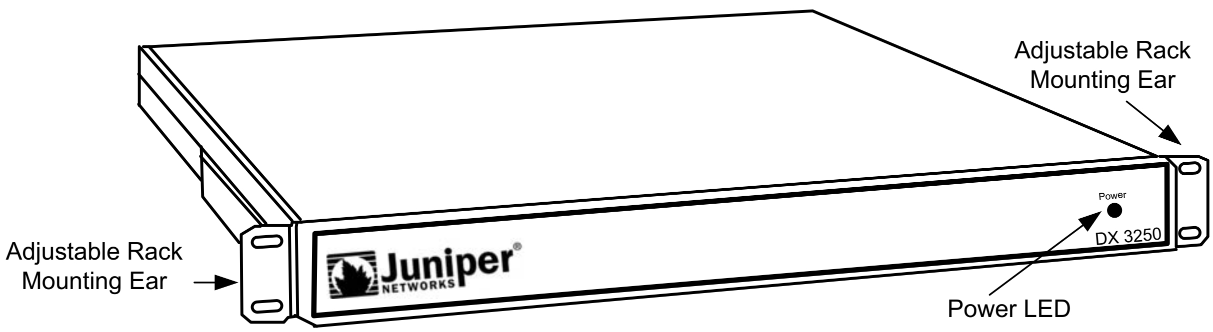


Figure 2 shows a rear view of the DX appliance 1U Chassis.

Figure 2: Rear View of the DX Appliance 1U Chassis

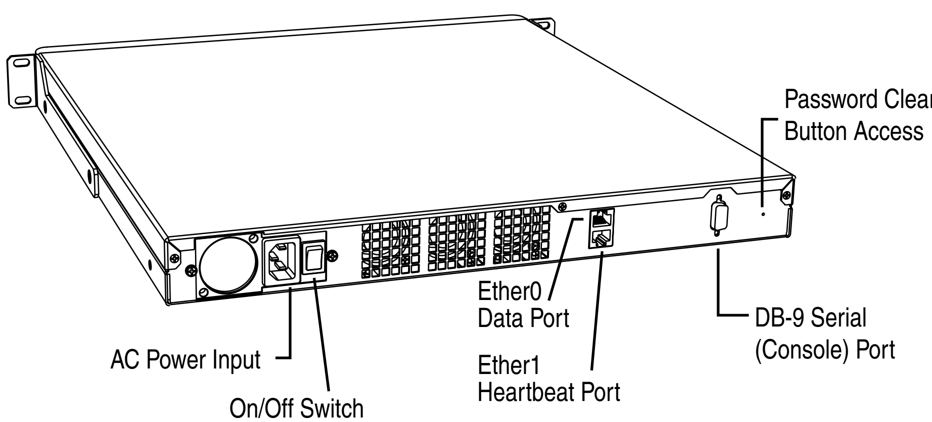


Figure 3 shows a front view of the DX appliance 2U Chassis.

Figure 3: Front View of the DX Appliance 2U Chassis

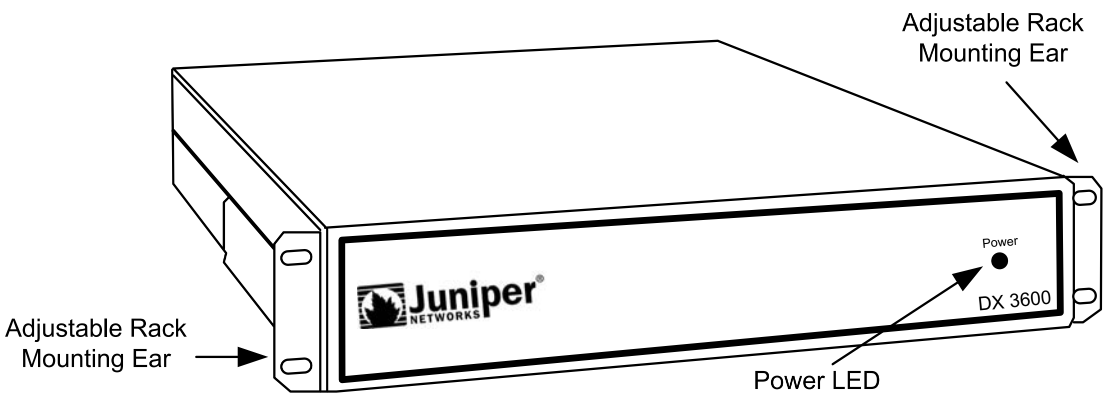


Figure 4 shows a rear view of the DX appliance 2U Chassis with four 10/100/1000BaseT Ethernet ports, and Figure 5 shows a rear view of the DX appliance 2U Chassis with two 10/100/1000BaseT Ethernet ports, and two fiber Gigabit Ethernet ports.

Figure 4: Rear View of the DX Appliance 2U Chassis with Four 10/100/1000BaseT Ethernet Ports

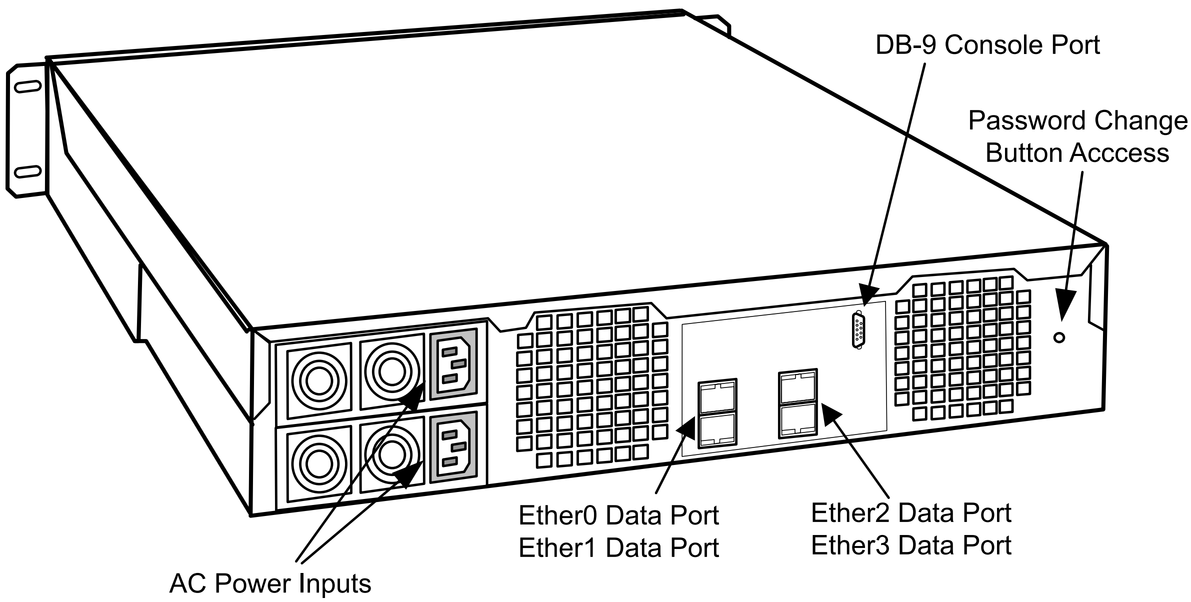
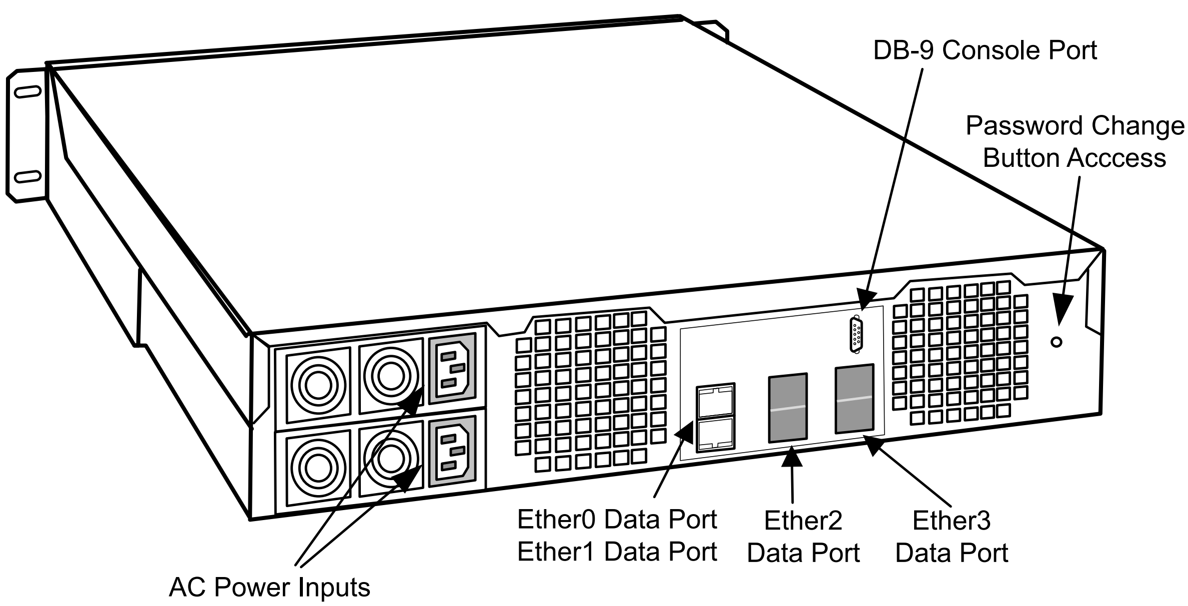


Figure 5: Rear View of the DX Appliance 2U Chassis with Two 10/100/1000BaseT Ports and two Fiber Gigabit Ethernet Ports.



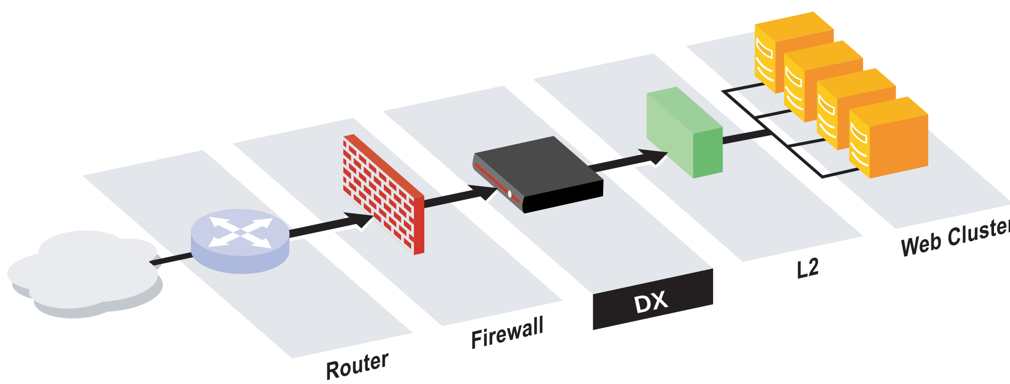
Terminology

To help you understand how to install and configure the DX appliance in your network, this section explains some of the more commonly used terms in this manual. For a complete list of terms, refer to “Glossary” on page 365. For additional examples of network topologies, refer to “Sample Network Topologies” on page 74.

Web Cluster

A Web Cluster (Figure 6) is a set of web servers to be accelerated. The DX appliance listens for incoming web traffic on a specific Virtual IP address and port, distributes it over the target hosts (web servers) in the cluster and then accelerates the outgoing web traffic. Typically all the web servers in a particular cluster serve identical content; that is, each cluster usually represents a distinct website or property.

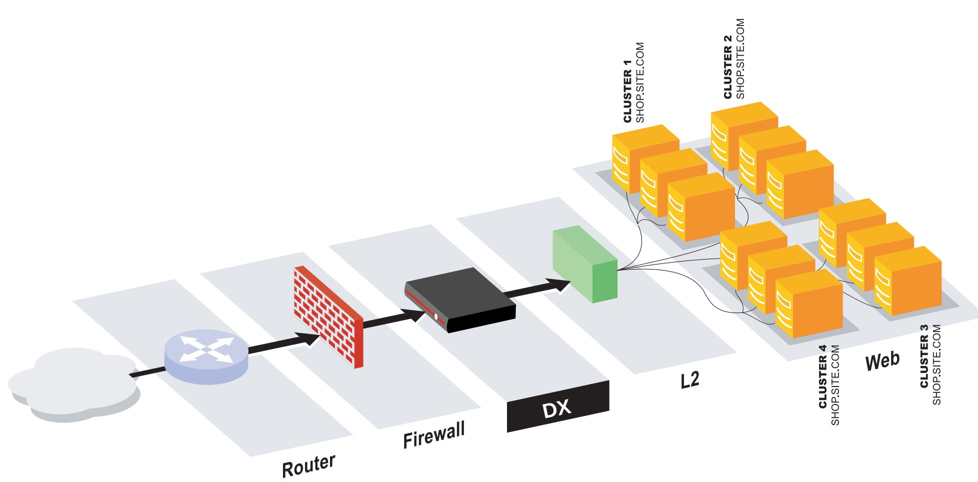
Figure 6: Examples of a Web Cluster



Web Farm

A Web Farm (Figure 7) is set of web clusters, typically with each cluster serving a different purpose or representing a separate website.

Figure 7: Examples of a Web Farm



Forwarder

A Forwarder is a DX appliance set up to forward traffic on to a set of servers without accelerating it. The DX appliance listens for incoming traffic on a specific virtual IP address and port, and then blindly distributes it to the appropriate target hosts. The hosts are typically not web servers, and the forwarder does not attempt to accelerate the outgoing traffic. This is for non-HTTP traffic; the forwarder simply passes the traffic through without examining it.

Redirector

A Redirector is a DX appliance set up to redirect requests to a single web server. It listens for incoming web requests on a specific virtual IP address and port and redirects the client to that web server. A redirector does not allow web traffic to pass through the Web I/O Accelerator. Instead, for every web request a redirector receives, the redirector sends the client back a redirect URL and forces it to resend its HTTP request directly to that URL.

The URL that the redirector sends back is composed of three portions:

- Redirector Host: The host portion of the redirector URL sent by the redirector. That is, this is the web server to which the client should be redirected. The redirector host may be specified as either a hostname or an IP address.
- Redirector Port: The port portion of the redirector URL sent by the redirector.
- Redirector Protocol: The protocol portion of the redirect URL sent by the redirector. Valid values are HTTP and HTTPS.

The manner by which the redirector specifies the path portion of the redirect URL is called the "Redirector URL" method. If the "Request" method is selected, then the redirector constructs the redirector URL using the same URL path as the original request. If the "Custom" method is selected, then the redirector constructs the redirector URL using a custom URL path. You must specify a custom URL path if the custom method is selected, and the custom URL path must begin with a slash "/".

For instance, if the request method is selected and the redirector receives a request for a page at "/path/page.html", then the redirector URL will look something like "http://my.redirect.host/path/page.html". However, if the custom method is selected and the custom URL path is set to "/custom/script.cgi?a=b", then the redirector URL will look something like "http://my.redirect.host/custom/script.cgi?a=b" for any request received by the redirector.

Chapter 2

First Time Configuration

This chapter describes the First Time Configuration process for the DX Application Acceleration Platform, discussing the following topics:

Information Required for First-Time Configuration

First-time configuration of the DX appliance requires the following information. Table 1 shows the information required for first-time configuration of the DX appliance.

Table 1: Information Required for First-Time Configuration

Required Information	Example
IP Address The IP address for the Remote Administration Interface port (Ether 0) for this DX appliance. This can be any arbitrary valid IP address on your subnet.	192.168.4.76
Netmask The Netmask (subnet mask) of this DX appliance.	255.255.0.0
Fully-qualified domain name The name of this DX appliance that will be set in DNS records.	dx.juniper.net
DNS Domain The DNS Domain (sometimes known as the DNS suffix) where this DX appliance is installed.	juniper.net
Primary Nameserver The Primary DNS server for this DX appliance.	192.168.0.5
Default route The Default route (sometimes called the gateway) for this DX appliance.	192.168.0.1
IP and Port of a Target Host for Cluster 1 The IP address of the server(s) you want to accelerate. Be sure to include the port number; for web servers this is usually port 80.	192.168.0.102:80
Fully-qualified Host Name for Cluster 1 This is full name of the web server(s) that the DX appliance will be accelerating -- the name clients use to reach this/these web server(s). This should equal the VIP in the DNS entry.	www.juniper.net
Virtual IP (VIP) Address for Cluster 1 If the DX appliance is not deployed behind other network devices such as a firewall or a server load balancer, the VIP should be assigned the publicly advertised address. Otherwise, the VIP can be assigned an arbitrary valid IP address on your subnet. This is the IP address that incoming internet traffic will be directed to and should be different from the IP addresses provided for Ether 0.	192.168.4.145
Username The username for this DX appliance. The default username is admin.	admin
Password The password for this DX appliance. The default password is admin.	admin

Connect a Terminal to the Console Port on the DX Appliance

You will need the null modem cable included with the DX appliance and any standard (RS-232) terminal or terminal emulator software (such as Windows HyperTerminal or SecureCRT) running on a PC.

NOTE: Because it is sometimes difficult to reach the DX appliance console port once it is mounted, consider completing the first-time configuration before mounting your DX appliance into an equipment rack or server cabinet.

1. Connect the (supplied) null-modem cable to the serial console port on the rear of the unit.

2. Connect the other end of the null- modem cable to the COM 1 port of a PC running a terminal emulator such as Windows HyperTerminal or SecureCRT (SecureCRT is available from VanDyke Software at www.vandyke.com).

Connect the DX Appliance to Your Network

Connect the DX appliance's primary Ethernet interface (Ether 0) to your network using a standard Ethernet cable.

CAUTION: For 1U units with Fast Ethernet (10/100/1000BaseT) ports, the DX appliance must be connected to a 10/100/1000BaseT full-duplex network port. The media settings on your switch for the port where the DX appliance is connected must match those for the DX appliance exactly.

CAUTION: For 2U units with Gigabit Ethernet ports (fiber), the DX appliance must be connected to a Gigabit switch with the media settings configured to autoselect.

NOTE: The Heartbeat interface, Ether 1, does not need to be connected to your network if you are installing a standalone DX appliance without a second DX appliance unit as a failover unit.

Power-up the DX Appliance

1U DX Appliance Models

1. Connect the supplied power cord to the power supply on the back of the DX appliance.

2. Flip the power switch to the "on" position. The LED on the front of the DX appliance will glow when the DX appliance has power, and the LED on the power supply will glow green.

3. It may take the DX appliance several minutes to boot.

2U DX Appliance Models with Dual Power Supply

1. Connect the supplied power cord to the power supply on the back of the DX appliance. The LED on the front of the DX appliance will glow brightly when the DX appliance has power, and the LED on the power supply will glow green.

NOTE: The DX appliance's dual power supply has no power switch. Connecting a hot power cord to the DX appliance will turn it on and begin the boot process.

NOTE: The power supply will emit a long startup beep if there is no power to the second power supply. Pressing the red buzzer reset button to the left of the plug will terminate the beep. This is normal.

2. It may take the DX appliance up to two minutes to boot; allow several minutes before proceeding.

Connecting to the DX Appliance with a Terminal or Terminal Emulator

If you are using a terminal emulator, be sure that the emulator is configured with the settings listed as:

- Bits per second: 9600
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none

You must create a connection to use Windows Hyper Terminal. Your first configuration screen should look like the one shown in Figure 8.

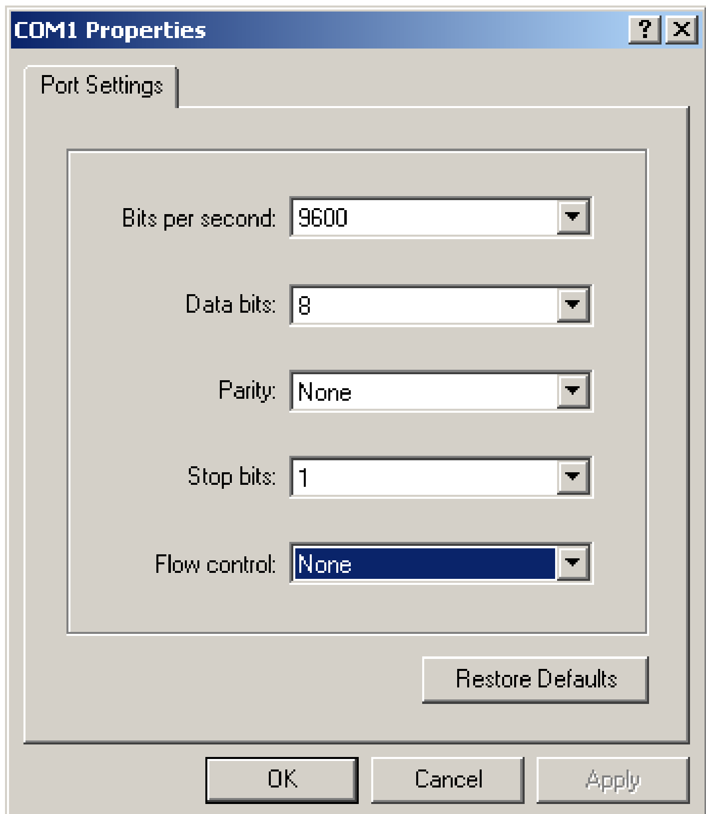
Figure 8: Hyper Terminal Connection Description Dialog Box

1. Enter a name that will be easy to identify. Hyper Terminal will then ask you which serial port you will be using as shown in Figure 9.

Figure 9: Hyper Terminal Connection Dialog Box

2. The last step in creating a connection is to configure the communication port properties as shown in Figure 10. Configure the communication parameters as shown.

Figure 10: Hyper Terminal Port Configuration Dialog Box



NOTE: If you are using Windows Hyper Terminal, after clicking the OK button to confirm your settings in the dialog box as shown in Figure 10, you may still need to click the CALL button or select CALL from the CALL pull-down menu to establish the connection.

---

3. Open the terminal connection to the DX appliance and press ENTER to log-in. You will see a screen similar to the one shown in Figure 11.

Figure 11: The DX Appliance First-Time Configuration Screen

If you do not see the screen as shown in Figure 11 and are unable to open a connection to the DX appliance:

1. Be sure that you have given the DX appliance enough time to boot up.

2. If you are using Windows Hyper Terminal, be sure to use CALL to establish a connection after entering the terminal settings. Even if it says CONNECTED in the lower left-hand corner of the Hyper Terminal window, you may not be connected until you use CALL.

3. Try pressing ENTER again to log-in.

4. Double-check that the null modem cable is connected to the COM 1 port of the PC.

5. Double-check that your terminal emulator is configured as previously described.

Logging-In for the First Time

If you have not previously set the username and password for the DX appliance, they will be set to their default values:

- Username: `admin`
- Password: `admin`

Log into DX appliance using the appropriate username and password. Continue onto the next step once you have logged-in.

Read and Agree to the License Agreement

Before you can continue with first-time configuration, you must agree to the License Agreement that appears when you first boot the DX appliance. Use the space bar to display each page of the License Agreement until you reach the end. When prompted, type `yes` and press the ENTER key.

Answer the Configuration Questions

The Juniper Networks First-Time Configuration program utility will ask you to provide values for twelve (12) basic configuration parameters required (refer to “Information Required for First-Time Configuration” on page 8) to get the DX appliance up and running in your network.

Table 1 on page 8 shows the questions that the DX appliance will ask you along with an explanation of each item. Items shown in brackets (e.g., [172.17.0.2]) are the factory defaults provided to serve as examples for your input. You must provide valid settings for the DX appliance to function in your network. Omit the brackets ([ ]) when typing your input.

CAUTION: If you make a mistake as you go through the first-time configuration, press CTRL-C and then press ENTER to quit. Then, to re-enter the first-time configuration program, type the command `config` at the DXSHELL prompt and press ENTER.

Table 2: Questions from the DX appliance First-Time Configuration Utility

First-time Configuration Questions
IP Address [172.17.0.2]:
Set the IP address of this DX appliance.
Netmask [255.255.255.0]:
Set the Netmask (subnet mask) of this DX appliance.
Fully-qualified domain name [juniper.juniper.net]:
Set the public name of this DX appliance that will be set in DNS records.
DNS Domain [juniper.net]:
Set the DNS Domain (domain suffix) where this DX appliance is installed.
Primary Nameserver [192.168.0.2]:
Set the Primary Nameserver for this DX appliance.
Default route [172.17.0.1]:
Set the Default route (gateway) for this DX appliance.
IP and port of a target host for cluster 1 (or <Enter> when done):
Enter the IP address of the server(s) you want to accelerate. Be sure to include the port number (for web servers this is usually port 80).
Fully-qualified host name for cluster 1 [www.yourdomain.com]:
Enter the server name of the webserver(s) that the DX appliance will be accelerating (the name that clients normally use to reach this/these webserver(s). This should be the VIP address that is in the DNS.
Virtual IP (vip) Address for cluster 1 [172.17.0.3]:
Set the IP address where incoming internet traffic will be directed. It must be the different from the IP address specified in question 2 and the same VIP as that of the DX appliance failover unit.
Do you want to run the Web Administration Server? [N]:
Typing Y will allow you to monitor and configure the DX appliance through a web browser by entering the address of the DX appliance and the default Web Admin Port 8090 in your browser (e.g., http://192.168.0.168:8090).[1] Access to the Web Administration Manager is password protected and can be turned off at any time.
Do you want to allow administration access via ssh? [Y]:
Type Y for Yes or N for No. Typing Y will allow you to monitor and configure the DX appliance through a secure Secure Shell (SSH) terminal session. This can be turned off at any time.
Do you want to allow administration access via telnet? [N]:
Type Y for Yes or N for No. Typing Y will allow you to monitor and configure the DX appliance remotely via telnet. This can be turned off at any time.

1. It is possible to configure the WebUI administrator to listen on an IP (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

After answering all the first time configuration questions as shown in Table 2, you are finished configuring the DX appliance. You should see the following message:

```
Configuration complete.
Writing configuration.
Done.
dx%
```

You are at the DXSHELL command line. The DXSHELL prompt will display the hostname that you assigned to the DX appliance using the first-time configuration utility, followed by the "%" sign (dx% in our example) the next time that you log in.

More information on configuring particular aspects of the DX appliance is presented in the chapters that follow.

Changing the Default Administrator Account Password

For security reasons, as soon as you have configured your DX appliance, you should immediately change the password for the default administrator "admin". Instructions for doing this are shown in sections, "Managing Users" on page 39 and "Changing a User's Attributes" on page 41. If, for any reason you cannot log onto any of the administrator accounts, you can reset the "admin" administrator password to its default value using the procedure described in "Deleting all Users and Resetting the Password for the User "admin"" on page 36.

Chapter 3

Remote Administration Interfaces

This chapter describes the Remote Administration Interface process for the DX Application Acceleration Platform, discussing the following topics:

Overview

The DX Application Acceleration Platform provides a variety of administrative interfaces to suit your environment and security needs. The DX appliance command line interface, DXSHELL, contains a comprehensive set of commands that allow you to view and change every aspect of the DX appliance configuration. For a list of all commands, refer to the *Command Line Reference* manual.

DXSHELL can be configured for access via:

- Secure Shell (SSH)
- Telnet
- Direct serial connection to the DX appliance console port

The browser-based WebUI provides access to the most frequently used configuration options. The WebUI can be configured for access via:

- Web browser
- Web browser with Secure Socket Layer (SSL) encryption

The DX appliance includes a custom Management Information Base (MIB) that allows you to view the DX appliance's configuration and status via SNMP. The SNMP agent also sends generic traps and enterprise-specific traps.

The Command Line Interface

The DX appliance command line interface, DXSHELL, can be accessed by:

- SSH
- Telnet
- Direct serial connection to the DX appliance console port

All three methods provide identical access to the DXSHELL command line interface.

Using SSH to Access the DX Appliance Command Line

The DX appliance can be accessed through a Secure Shell (SSH) client. Using SSH ensures that while you are connected to the DX appliance, all information passing between you and the DX appliance is encrypted for security. You will need to have an SSH client or application installed and functioning on the computer from which you will access the DX appliance.

1. If you are using a command line SSH client, type the following command:

   ssh admin@<IP address of DX appliance>

   If you are using a PC with a terminal emulator application that supports SSH, configure it to connect to the IP address of the DX appliance. When you are

prompted for the username, either enter "admin" for the default account, or the name of a user account that you have created.

2. You will be prompted for a password. Enter the password for the DX appliance.

   You will see the `%` prompt that indicates that you have reached the Juniper Networks DXSHELL, a custom command-line interface.

3. Type `help` or press the tab key to see a list of commands. You can also refer to the *Command Line Reference* manual for a list of all the DXSHELL commands and their descriptions.

4. You can disconnect from DXSHELL at any time by entering the command:

   ```
   dx% exit
   ```

or

   ```
   dx% quit
   ```

Using Telnet to Access the DX Appliance Command Line

The DX appliance can be accessed through a standard Telnet client. You will need to have a Telnet client or application installed and functioning on the computer from which you will access the Juniper Networks DX appliance.

NOTE: The DX appliance's Telnet administration service must be turned-on in order to connect to the DX appliance through the Telnet.

---

1. If you are using a command line Telnet client, type the following command:

   ```
   telnet <IP address of DX appliance>
   ```

2. If you are using a PC with a terminal emulator application, configure the emulator to connect to the IP address of the DX appliance.

3. You will be prompted for a username and password. Enter the username and password that you set for the DX appliance.

4. You will see the `%` prompt that indicates that you have reached the Juniper Networks DXSHELL, a custom command-line interface.

5. Type help or press the tab key to see a list of commands. You can also refer to the *Command Line Reference* manual for a list of all the DXSHELL commands and their descriptions.

6. You can disconnect from the DXSHELL at any time by entering the command:

   ```
   dx% exit
   ```

or

   ```
   dx% quit
   ```

Using a Console Port to Access the DX Appliance Command Line

The DX appliance can be accessed through a direct serial connection to the console port on the back of the unit. The console connection must be used for the first-time configuration. After that, it provides out-of-band management capability.

1. Connect one end of the supplied null modem cable to the serial (console) port on the rear of the unit.

2. Connect the other end of the cable to the COM1 port of a PC running terminal emulation software or any standard RS-232 terminal. Use 9600 baud, 8 bits, no parity (refer to "Connect a Terminal to the Console Port on the DX Appliance" on page 9 for details).

3. Open the terminal session and press ENTER to bring up communication with the DX appliance.

4. You will be prompted for a username and a password. Enter the username and password for the DX appliance. If this is the first time that you have logged in, use the default account with the username "admin" and the password "admin". You will see the `%` prompt that indicates that you have reached the Juniper Networks DXSHELL, a custom command-line interface.

5. Type help or press the tab key to see a list of commands. You can also see the *Command Line Reference* manual for a list of all the DXSHELL commands and their descriptions.

6. You can disconnect from the DXSHELL at any time by entering the command:

```
dx% exit
```

or

```
dx% quit
```

Making Changes from the Command Line

The commands `show` and `set` are used to view and change all the configurable parameters for the DX appliance. A complete list of parameters accessible with the `set` command, along with examples, is provided in the *Command Line Reference* manual.

CAUTION: After using the `set` and `clear` commands to make changes you will see the (*) prefix at the command line prompt. This indicates that configuration settings have been changed, but the changes have not yet been saved. With the exception of a few commands, changes do not take effect and are not saved until you enter the `write` command.

To apply and save the configuration changes, enter the command:

```
dx% write
```

If you have not yet entered the `write` command, you can revert to the configuration settings that existed before changes were made by entering the command:

```
dx% reload
```

`Set` commands that control the state of the DX appliance unit take effect immediately without use of the `write` command, as follows:

- `set server [ up | down ]`
- `set admin ssh [ up | down ]`
- `set admin telnet [ up | down ]`
- `set admin webui [ up | down ]`
- `set admin snmp [ up | down ]`

NOTE: If you wish to preserve the configuration so that it becomes active again on the next boot-up, you must follow these `set` command with a `write` command.

Command Abbreviation

The Command Abbreviation feature allows you to type abbreviated DXSHELL commands that are then resolved and executed by the DX appliance. The output delivered by the execution of unambiguous commands will be the same as its non-abbreviated command equivalent. However, if a command is ambiguous, that will result in an error string such as “Ambiguous Keyword”. The DX appliance will also suggest possible matches:

```
dx% cl cluster
Ambiguous keyword: “cl”
Possible matches:
clear
cls
dx%
```

For example, the DXSHELL command used to check health interval for Cluster 1 is:

```
dx% show cluster 1 health interval
```

The abbreviated command equivalent is:

```
dx% sh clu 1 he in
```

Command abbreviation is subject to these restrictions:

- Both commands and parameters can be abbreviated. For example, you can abbreviate the show command to `sh` because show is the only command that begins with `sh`.

- The abbreviation must contain enough letters to differentiate it from the other commands and parameters at that level. For example, `sh cl` is not unique in its parameter so you must type `sh clu` to specify show cluster. In addition, the command `sh clu 1 st` is not a unique command as there are two possible interpretations: `show cluster 1 stats` and `show cluster 1 sticky`. A unique abbreviated command for `show cluster 1 stats` would be `sh clu 1 sta`.

- The determination of a “unique” command or parameter of a command is made dynamically. User-defined names (a cluster name, for example) are not considered part of the command syntax check, and a command that can be resolved by the system without considering user-defined names or strings will be executed.

The Web User Interface (WebUI)

The WebUI provides access to the most commonly used DX appliance configuration parameters in a familiar and easily-accessible web interface. Users with the Administrator role have read-write access to all pages on the WebUI. Users with all other access roles have read-only access to the WebUI pages. This includes users with access roles network_administrator, network_operator, security_administrator, or user (refer to "Multi-Level Administration Rights" on page 35).

NOTE: Using the WebUI requires that you have at least Netscape version 6.x, Internet Explorer version 5.x or Opera version 6.x installed. Earlier versions of these browsers will not work with the WebUI.

Turning on the WebUI

If you did not enable the WebUI Server during initial configuration, or if it is not otherwise available, you will have to access the DXSHELL command line to turn it on.

1. Access DXSHELL either through a direct terminal connection or remotely using SSH or Telnet. The default port for the WebUI is 8090.

2. From the DXSHELL command-line interface, enter the following commands:

```
dx% set admin webui port <number>
dx% set admin webui up
dx% write
```

Setting the WebUI Interface to Communicate over the SSL

If you plan on accessing the WebUI over an unsecured connection, you should enable "Secure Socket Layers" for the WebUI. This is an optional process that should only be used when extra security is needed.

1. Access DXSHELL either through a direct terminal connection or remotely using SSH or Telnet. This may already be in place if you are continuing from the previous steps.

2. From the DXSHELL command-line interface, enter the following commands:

```
dx% set admin webui ssl keyfile demokey
dx% set admin webui ssl keypass
dx% set admin webui ssl certfile democert
dx% set admin webui ssl enabled
dx% write
Writing configuration.
Done.
```

   This example uses the dummy key and certificate files named demokey and democert, respectively. If you are installing the DX appliance in a production environment, make sure you have valid key and certificate files in base-64 encoding. Instructions for importing these files from a variety of environments, as well as converting them to base-64, appear in "Importing Existing Keys and Certificates" on page 174.

When importing key files from different environments, occasionally they will need to be converted using the OpenSSL software. For information on this program, refer to the open SSL web pages at:

http://www.openssl.org/

3. To see the current WebUI SSL setup, type the command:

   `dx%` **`show admin webui`**

   The DX appliance will respond with the current setup:

```
Port: 8090
SSL Status: enabled
SSL Keyfile: demokey
SSL Keypass: none
SSL Certfile: democert
Session Expire Time: 900
Web UI: up
```

Accessing the WebUI

1. Open a web browser (you may need to be inside your company's firewall to access the web interface).

2. Type the DX appliance host name or IP address along with the port on which the WebUI is listening (the default port is 8090) in your browser's address bar. The URL may look something like this:

   http://192.168.100.100:8090

or

   http://dx.yourdomain.com:8090

3. You will be prompted to enter your username and password. Use the default username `admin` and password `admin` or one of the previously-defined user accounts. The password is the one that you set during the first-time configuration.

NOTE: It is possible to configure WebUI administrator to listen on an IP address (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

---

4. After you enter your username and password, the WebUI “Dashboard.” will be displayed. Use the dashboard to configure your DX appliance.

Logging out of the WebUI

When you have finished your WebUI session, you should log out of your administration session using the Logout button. Then close the browser window and quit your web browser to prevent anyone from re-opening your WebUI session. This prevents someone from using that browser to access the WebUI.

If you forget to log out, the session will automatically time out after a fixed period of time. You will then have to log in again before you are able to access the WebUI.

Working with the WebUI

After logging in, you will see the WebUI dashboard (Figure 12). From the dashboard, you can view and change all of the DX appliance's network settings, and you can access other settings pages from the navigation menu on the left-hand side of the page. The following links appear in the left-hand menu. Click on the links to view the associated settings:

- Dashboard
- Clusters, Forwarders, and Redirectors
- DX appliance Statistics
- Cluster Statistics
- Network Settings
- Date & Time
- Admin Services
- Users
- Audit Log
- SNMP
- Keys and Certificates
- Support

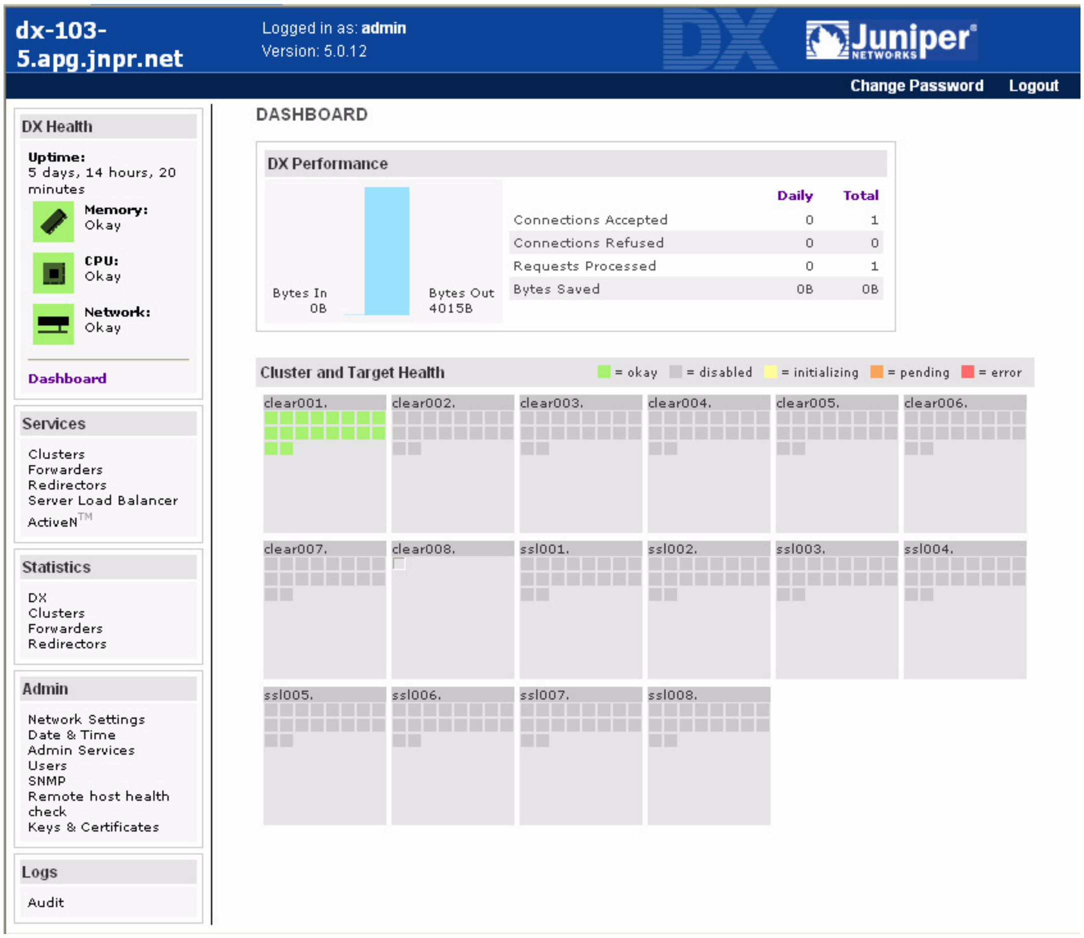
Making Changes with the WebUI

The WebUI lets you view and change settings with a familiar forms-based web interface. To make changes:

1. Select the desired option or enter the desired value.
2. Click the SAVE button at the bottom of the page. Your changes will be saved and applied immediately.

If you make a mistake and do not want to save your changes, you can click your browser's refresh button to get a fresh copy of the page. You can also select one of the other settings pages from the left-hand navigation menu and your changes will not be saved.

Figure 12: The WebUI Dashboard



On-Line Help in the WebUI

On-line help is available by clicking on the terms that appear next to each field. When you click on a term, a pop-up window will open and the term and its definition will appear at the very top of the window.

SNMP Agent

Overview of the SNMP Agent

The SNMP agent supports SNMP Version 2c for SNMP get and getnext and version 1 and 2c for SNMP traps. The SNMP agent does not support the SNMP set operation. Security is provided through SNMP community strings. The default community strings are “public” for the SNMP getnext operation. The community strings can be modified through either DXSHELL or the WebUI. SNMP traps do not have a default setting; you must configure a trap.

Juniper Networks is registered as Enterprise 6213. Detailed SNMP Management Information Base (MIB) and trap definitions for the SNMP agent can be found in the following Juniper Networks Enterprise MIB documents:

- DX-MIB: Juniper enterprise top level MIB definitions
- DX-CONFIG-MIB: Juniper enterprise configuration MIB definitions
- DX-STATS-MIB: Juniper enterprise statistics MIB definitions
- DX-TRAP-MIB: Juniper enterprise trap definitions (SNMP v1.0)
- DX-TRAPv2-MIB: Juniper enterprise trap definitions (SNMP v2.0)

Users may specify up to two trap hosts for receiving SNMP traps. The agent will send the SNMP trap to the specified hosts when appropriate. The SNMP agent can send version 1 and version 2 traps formats. Traps will not be sent when there is no host specified.

The SNMP agent supports the standard MIB, RFC 1213- MIB II, and the following generic traps:

- ColdStart
- WarmStart
- LinkDown
- LinkUp
- Authentication Failure

In addition, it supports the Enterprise SMNP traps shown in Table 3.

Table 3: Enterprise SNMP Traps Supported

Trap Name	Description
failoverStateActive	Indicates that the Juniper Accelerator is assuming the active role.
connectionThresholdTrap	Indicates that the Juniper Accelerator has reached the threshold for the maximum number of connections on the client side.
TargetServerStateUp	Indicates that the target server is up.
TargetServerStateDown	Indicates that the target server is down.
vipStateDown	Indicates that the VIP is down.
vipStateUp	Indicates that the VIP is up.

Configuring the SNMP Agent Parameters

The following steps are used to set up the SNMP agent:

1. Enable the SNMP service by typing:

   ```
   dx% set admin snmp up
   ```

2. Define the System location by typing:

   ```
   dx% set admin snmp location <location>
   ```

or

   ```
   dx% set admin snmp location snmp q/a lab, rack 4
   ```

3. Define the System contact by typing:

   ```
   dx% set admin snmp contact <contact>
   ```

or

   ```
   dx% set admin snmp contact John Smith
   ```

Configuring the SNMP Agent for Sending Traps

The following steps are used to set up the SNMP agent to send traps:

1. Define the trap host by typing:

   ```
   dx% set admin snmp trap host [1 | 2] ip <ip address>
   ```

or

   ```
   dx% set admin snmp trap host 1 ip 205.178.13.100
   ```

2. Define the community string for the trap host by typing:

```
dx% set admin snmp trap host [1 | 2] community <community string>
```

or

dx% **set admin snmp trap host 1 community my_community**

3. Define the SNMP version for the trap host. The agent supports both version 1 and version 2 formats.

   dx% **set admin snmp trap host [1 | 2] version [1 | 2]**

or

   dx% **set admin snmp trap host 1 version 1**

4. Enable sending of generic traps by typing:

   dx% **set admin snmp trap generic [enabled | disabled]**

or

   dx% **set admin snmp trap generic enabled**

5. Enable sending of Enterprise-specific traps by typing:

   dx% **set admin snmp trap enterprise [enabled | disabled]**

or

   dx% **set admin snmp trap enterprise enabled**

6. OPTIONAL. Enable or disable sending of Authentication Failure traps by typing:

   dx% **set admin snmp trap authfailure [enabled | disabled]**

or

   dx% **set admin snmp trap authfailure enabled**

7. OPTIONAL. Define the threshold for connections count by typing:

   dx% **set admin snmp trap threshold connection <1-100%>**

or

   dx% **set admin snmp trap threshold connection 95**

Administrator Remote Authentication

Administrator Remote Authentication allows a properly-enabled administrator to log onto and administer the DX appliance using the Command Line Interface (CLI) from anywhere in the world. The connection uses a secure protocol (DAP and RADIUS) for remote authentication.

There are two classes of users (administrators of the DX appliance): local and remote. By default, when a new user is added, his class is set to local. The class of a user is set to remote using the DXSHELL command:

```
dx% set user <user> class <local | remote>
```

Remote authentication is only performed for users whose class is remote.

For all the users, the assigned roles are stored locally on the DX appliance. Because of this, all users, local or remote, have to be added on the DX appliance. For a remote user, a password does not have to be set on the DX appliance because the authentication is handled by the authentication server.

The default role for the remote users is “user.” If no role is specifically set for a remote user, the default role is used. The default role can be changed using the command:

```
dx% set admin remoteauth userrole <role>
```

The login class is used to differentiate between a local and a remote user. Currently LDAP and RADIUS are supported for remote authentication; the default protocol is RADIUS. It can be changed using the command:

```
dx% set admin remoteauth protocol <ldap | radius>
```

When a user tries to login, through the console, telnet, SSH, and WebUI, the DX appliance uses this logic to authenticate:

- If the user is a local user, authentication takes place locally as before.
- If the user is a remote user:
  - If remote authentication is not enabled, the user login is refused
  - If the required current protocol (LDAP or RADIUS) configuration is not present, the user login is refused.
  - If remote authentication is enabled and all of the required current protocol (LDAP or RADIUS) configuration is present:
  - LDAP or RADIUS server 1is contacted for authentication
  - If there is a communication error with server 1 or server 2 is contacted for authentication.
  - If authentication does not succeed, user login is refused.

- If authentication succeeds:
  - A remote authorization user role is assigned to the user.
  - If not, the default role is assigned.

Remote Authentication Configuration Commands

These commands are used to configure Administrator Remote Authentication.

Set Commands

To enable or disable Administrator Remote Authentication, type the command:

```
dx% set admin remoteauth status <enabled | disabled>
```

To set the authentication protocol to use for Administrator Remote Authentication, type the command:

```
dx% set admin remoteauth protocol <ldap | radius>
```

To set the default role for remote users, type the command:

```
dx% set admin remoteauth userrole <role>
```

The default role is “user”.

To set the class attribute of a user, type the command:

```
dx% set user <user> class <local|remote>
```

To set the RADIUS server password, type the command:

```
dx% set admin remoteauth radius server key <key>
```

To set the IP address for RADIUS server 1, type the command:

```
dx% set admin remoteauth radius server 1 ip <ip>
```

To set the port for RADIUS server 1, type the command:

```
dx% set admin remoteauth radius server 1 port <port>
```

To set the IP address for RADIUS server 2, type the command:

```
dx% set admin remoteauth radius server 2 ip <ip>
```

To set the port for RADIUS server 2, type the command:

```
dx% set admin remoteauth radius server 2 port <port>
```

To set the Distinguished Name (DN) of the node in the LDAP Directory Information Tree, under which the users have to be searched, type the command:

```
dx% set admin remoteauth ldap basedn <base-dn>
```

To set the attribute name that uniquely identifies the user in LDAP database, type the command:

```
dx% set admin remoteauth ldap uid <uid>
```

To set the Distinguished Name of the LDAP admin user, type the command:

```
dx% set admin remoteauth ldap bind userdn <user-dn>
```

The DX appliance authenticates itself with the LDAP servers using this user DN.

To set the password for the LDAP admin user, type the command:

```
dx% set admin remoteauth ldap bind password <password>
```

To set the IP address for LDAP server 1, type the command:

```
dx% set admin remoteauth ldap server 1 ip <ip>
```

To set the port for LDAP server 1, type the command:

```
dx% set admin remoteauth ldap server 1 port <port>
```

To set the IP address for LDAP server 2, type the command:

```
dx% set admin remoteauth ldap server 2 ip <ip>
```

To set the port for LDAP server 2, type the command:

```
dx% set admin remoteauth ldap server 2 port <port>
```

Show Commands

To display the current status of remote authentication, type the command:

```
dx% show admin remoteauth status
```

To display the current remote authentication protocol, type the command:

```
dx% show admin remoteauth protocol
```

To display the default user role for remote users, type the command:

```
dx% show admin remoteauth userrole
```

To display the RADIUS server key, type the command:

```
dx% show admin remoteauth radius server key
```

To display the IP Address and port for RADIUS server 1, type the command:

```
dx% show admin remoteauth radius server 1
```

To display the IP Address and port for RADIUS server 2, type the command:

```
dx% show admin remoteauth radius server 2
```

To display the LDAP base-dn, type the command:

```
dx% show admin remoteauth ldap basedn
```

To display the LDAP uid, type the command:

```
dx% show admin remoteauth ldap uid
```

To display the LDAP admin user-dn, type the command:

```
dx% show admin remoteauth ldap bind userdn
```

To display the LDAP admin password, type the command:

```
dx% show admin remoteauth ldap bind password
```

To display the IP Address and port for the LDAP server 1, type the command:

```
dx% show admin remoteauth ldap server 1
```

To display the IP Address and port for the LDAP server 2, type the command:

```
dx% show admin remoteauth ldap server 2
```

Chapter 4

Multi-Level Administration Rights

This chapter provides an overview of Multi-Level Administration Rights for the DX Application Acceleration Platform, discussing the following topics:

Overview

To enable better management and user accountability, different levels or classes of user access have been implemented on the DX Application Acceleration Platform. The classes of users are called “Roles”. The level of access increases as needed to perform various management tasks. This allows you to differentiate:

- Users vs. Administrators vs. Operators
- Network administration vs. Security administration

Conceptually, roles can be grouped as follows:

- The user’s interaction with the DX appliance is completely passive, i.e., nothing can be changed on the DX appliance. Users can display information but can not make any configuration or operational state changes. This is useful for users in the Network Operations Center (NOC) that need to view information on all devices but not make any changes.
- Operators have access to the DX appliance management features used for daily operations. Operators should not be allowed to make configuration changes. Operators can only view information and enable/disable services and target servers. Operators should not be able to severely impact the operation of the DX appliance.
- Administrators are the only ones that may make permanent changes to the DX appliance configuration. Administrators can access all the functions to configure and troubleshoot problems on the DX appliance.

User Access Levels

A user can be assigned to one or more roles as defined in Table 4. Access to the DX appliance must be controlled by a unique username and password combination. Once you are connected via local console, Telnet, or SSH, you are prompted to enter a username and a password.

Default Account on the DX Application Acceleration Platform

The default account for the DX appliance is:

- Username: `admin`
- Password: `admin`
- Role: administrator (see below for description)

The first time you log into a DX appliance through the serial console port, you must log in with the default username and password. As part of the first time configuration procedure, you will have an option to change the password for the default account. It is recommended that you change the default password or disable the account after initial configuration. The default account cannot be deleted and the role cannot be changed.

If you upgrade to a newer version of the firmware from a 2.3.X or 3.0.X DX appliance, you will need to login using the default username and the same password that you had previously defined for the default username on the DX appliance before the installation of the new firmware.

Deleting all Users and Resetting the Password for the User "admin"

- Pressing the "PASSWORD RESET" button on the back of the DX appliance and holding it for four seconds will enable the default account and reset the password of that account. This action does not affect any other user accounts.

- You can delete all user accounts on the DX appliance by logging in as the default user (admin) and typing the following command. This command will not delete the default account.

  ```
  dx% delete user all
  ```

- To reset the DX appliance to factory default settings and delete all user accounts, except the default account, type the following commands while logged in as the default user.

  ```
  dx% delete user all
  dx% reset config
  ```

Table 4: Roles

Role	Description and Tasks Performed
administrator	The administrator has complete access to all DXSHELL commands on the DX appliance. Administrators may add new users and change user attributes.
network_administrator	The network_administrator can execute all DXSHELL commands, except those related to SSL.
network_operator	The network_operator can execute all DXSHELL commands that don't change the configurations and settings to the DX appliance, except those related to SSL. In addition, the network_operator can enable and disable the following: ■ Target Servers ■ State of services ■ Server ■ Telnet ■ Web Administration Server ■ SSH ■ SNMP
security_administrator	The security_administrator can execute all DXSHELL commands for SSL features only.
security_operator	The security_operator can view all SSL configuration and statistics, but cannot change the configuration related to those features.
user	The user can view all status information and statistics, except SSL related information, and cannot make any configuration changes or service state changes to the DX appliance. This is extremely useful for users in a NOC that can only view information on devices.
target_operator	The target host operator has the same capabilities as a user, but can also enable, disable, pause, or unpause a target host within a cluster.

Valid User Names and Passwords

- Usernames and passwords are case-sensitive
- Usernames must be between 4-16 characters long
- Passwords must be at least 6 characters long

Exporting and Importing User Accounts

You can export and import user account information for backup purposes or to match user accounts across multiple units. Exporting and importing requires:

- Access to the command line
- A Trivial File Transfer Protocol (TFTP) or Secure Copy (SCP) server

Exporting User Accounts

To export user accounts, enter the following command:

```
dx% export users tftp://<tftpservername>/<accountsfilename>
```

or

```
dx% export users scp://<scpservername>/<accountsfilename>
```

Exported Account Information

Exported user accounts information can help you match user accounts across multiple DX appliances. An exported user account file consists of the list of DXSHELL commands required to completely recreate the user accounts.

You can use a text editor to customize the account information, removing or commenting out commands. Lines beginning with a # are comments and will be ignored. You can use the following command to view the commands needed to recreate the user accounts:

```
dx% display users
```

Importing User Accounts

To import user accounts from a TFTP server, enter the following command:

```
dx% import users tftp://<tftpservername>/<accountsfilename>
```

or

```
dx% import users scp://<scpservername>/<accountsfilename>
```

Managing Users

Administrators are the only users that can add new users and change users' attributes.

Adding a New User

The following are the steps for setting up a new user.

1. To add a new user, type:

```
dx% add user
```

or

```
dx% add user <username>
```

2. Enter the new username: <username> (the DX appliance will prompt you for the username if it is not provided). The system response will be similar to this:

```
dx% add user fred
User fred has been added. Please perform the following
to complete the addition of this user:
- set a password
- enable the user
- assign a role (optional)
dx%
```

Before one or more roles are assigned to a new user, a new user will have very limited rights and can only access the following commands:

```
cls
exit
help
history
ping
set password
show cluster
show commands
show forwarder
show hostname
show loginbanner
show redirector
show server
show support
show ua
show version
who
whoami
```

3. Set the password for the new user by typing:

```
dx% set user <username> password
New password:
Enter the new password again:
Password changed for user <username>.
```

For example:

```
dx% set user fred password
New password:
Enter the new password again:
Password changed for user fred.
```

4. Enable the new user by typing:

```
dx% set user <username> enabled
```

For example:

```
dx% set user fred enabled
```

The DX appliance response will be similar to this:

```
dx% user fred is now enabled
```

A user cannot be enabled unless a password has been assigned. If you try to enable a user without assigning a password, you will see a response similar to this:

```
dx% set user fred enabled
Error: Cannot enable user fred because that user has no password.
dx%
```

5. Assign one or more roles to a user as shown:

```
dx% set user <username> role <role1 role2 …>
```

The role can be one of [administrator | security_ administrator | security_operator | network_ administrator | network_operator | user].

For example:

```
dx% set user fred role network_administrator
dx% set user fred role network_operator security_operator
```

In the second example, user "fred" will have access rights for both network_ operator and security_operator. This is useful in an organization where an operator has administration responsibilities for both the network and security services.

The system responds with a line for each role:

```
dx% set user fred role security_administrator security_operator
Role security_administrator has been added to user fred's permissions.
Role security_operator has been added to user fred's permissions.
dx%
```

6. To display a user's information, type:

```
dx% show user
Sample output of show user:
User        Status      Roles
----        ------      -----
fred        Enabled     (none)
tom         Enabled     administrator
```

```
dick        Disabled    network_operator
harry       Enabled     network_administrator
```

Changing a User's Attributes

Changing the User's Password

To change a user's password, type:

```
dx% set user <username> password
```

For example:

```
dx% set user fred password
```

The system response will be similar to this:

```
dx% set user fred password
New password:
Retype new password:
Password changed for user fred.
dx%
```

No characters are echoed on password input, and the user's name is displayed on the final confirmation.

Clearing a User's Role

To clear a user's role, type:

```
dx% clear user <username> role <role1 role2 …>
```

The role can be one of [administrator | security_ administrator | security_operator | network_ administrator | network_operator | user].

For example:

```
dx% clear user fred role security_administrator
```

The system response will be similar to this:

```
dx% clear user fred role network_administrator security_administrator
Role network_administrator has been removed from user fred's permissions.
Role security_administrator has been removed from user fred's permissions.
dx%
```

Deleting a User

To delete a user, type:

```
dx% delete user <username>
```

For example:

```
dx% delete user fred
```

The system response will be similar to this:

```
dx% delete user fred
Are you sure you want to delete user fred (y/n)? [y]
User fred deleted.
dx%
```

Actions that Affect All Users

The following commands are applicable to all user accounts, with these exceptions:

- The default account, i.e., the user with the username "`juniper`".
- The user with the administrator role that is making the changes.

Enabling and Disabling All Users

To enable or disable all users, type:

```
dx% set user all [enabled | disabled]
```

For example:

```
dx% set user all enabled
```

The system will respond with a message similar to this:

```
dx% set user all enabled
Are you sure you want to set all users to 'enabled' (y/n)? [y]
User fred is now enabled.
User tom is now enabled.
User dick is now enabled.
User harry is now enabled.
dx%
```

Assigning Roles to All Users

To assign one or more roles to all users. Each user will have one or more roles added to the current assigned roles.

```
dx% set user all role <role1 role2 …>
```

The role can be one of [administrator | security_ administrator | security_operator | network_ administrator | network_operator | user].

For example:

```
dx% set user all role security_administrator
```

The system will ask for confirmation and then display one line for each user and role:

```
dx% set user all role network_operator network_administrator
Are you sure you want to change all users' roles (y/n)? [y]
Role network_operator has been added to user fred's permissions.
Role network_administrator has been added to user fred's permissions.
Role network_operator has been added to user tom's permissions.
Role network_administrator has been added to user tom's permissions.
Role network_operator has been added to user dick's permissions.
Role network_administrator has been added to user dick's permissions.
Role network_operator has been added to user harry's permissions.
Role network_administrator has been added to user harry's permissions.
```

```
4 users changed.
dx%
```

Clearing All User Roles

To clear all users' role. Each user will have one or more roles removed from the current assigned roles.

```
dx% clear user all role <role1 role2 …>
```

The role can be one of [administrator | security_ administrator | security_operator | network_ administrator | network_operator | user].

For example:

```
dx% clear user all role security_administrator
```

The system will ask for confirmation and then display one line for each user and role:

```
dx% clear user all role network_operator network_administrator
Are you sure you want to change all users' roles (y/n)? [y]
Role network_operator has been removed from user fred's permissions.
Role network_operator has been removed from user tom's permissions.
Role network_administrator has been removed from user tom's permissions.
Role network_operator has been removed from user dick's permissions.
Role network_administrator has been removed from user dick's permissions.
Role network_operator has been removed from user harry's permissions.
Role network_administrator has been removed from user harry's permissions.
4 users changed.
dx%
```

Deleting all users

To delete all users, type:

```
dx% delete user all
Are you sure you want to delete all users (y/n)? [n]y
```

The system response will be similar to this:

```
dx% delete user all
Are you sure you want to delete all users (y/n)? [n]y
User fred deleted
User tom deleted.
User dick deleted.
User harry deleted.
dx%
```

Chapter 5

Common Administration Tasks

This chapter describes the Common Administration Tasks for the DX Application Acceleration Platform, discussing the following topics:

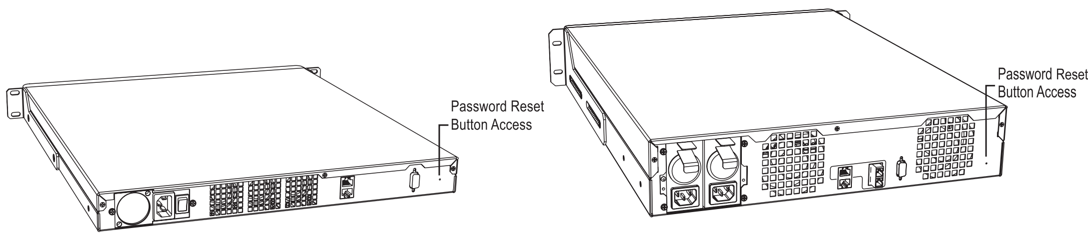
Overview

Dealing with a Lost Password

Resetting the DX Appliance Password

If you forget the password for your DX Application Acceleration Platform, you can reset the password to the default value using the PASSWORD RESET button accessible from the rear of the DX appliance. Refer to Figure 13.

Figure 13: Resetting the DX Appliance Password



Insert a paper clip into the password reset hole as shown in Figure 13. You will feel it come in contact with a button directly inside the box. If you press and hold the this button down for approximately four seconds, the following will occur:

- Any open administrative sessions will be closed
- SSH, Telnet, and WebUI access will be turned off
- The password for the default account will be reset to the default value.

None of the configuration settings will be changed. You will not need to repeat the First-Time Configuration Program. However, to set a new password you will need to connect to the DX appliance through a console port. You will be prompted for the default username (`admin`) and password (`admin`), and once you have entered these, you can set a new password with the command:

```
dx% set password
```

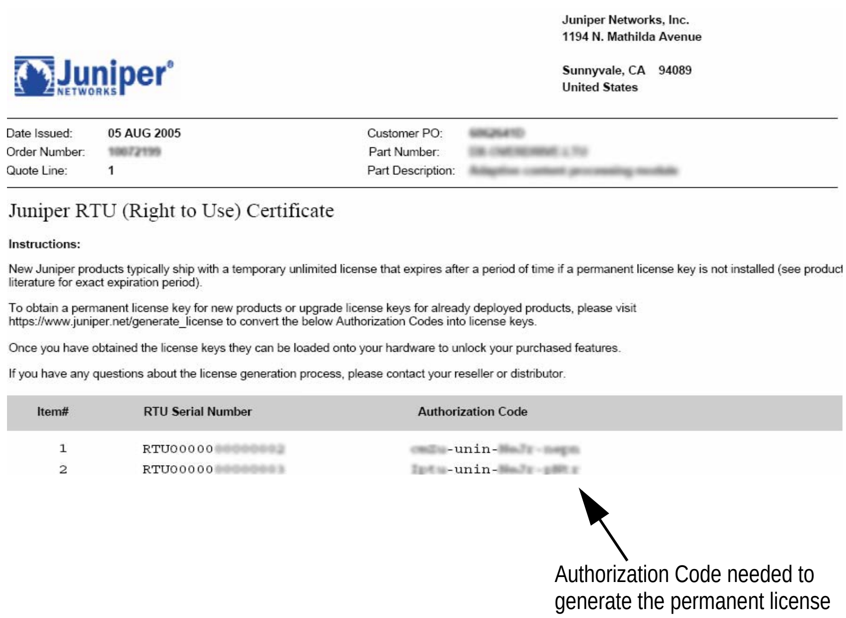
The License Key

New DX Application Acceleration Platforms have a built-in permanent license that allows access to all of the standard features.Only the Overdrive Application Rules (AppRules) and 3G Cache options need a separate license. To obtain a permanent license for these features, you need:

- A Juniper Customer Support Center (CSC) User ID and Password
- The device serial number (displayed on back of device)
- An Authorization Code used to generate the license for the WAN speed and each optional feature

The Authorization Code is provided in the *Juniper Right To Use* document that is E-Mailed to the address specified on the Purchase Order. (see Figure 14)

Figure 14: Example of the Juniper Right to Use Certificate



Juniper Networks, Inc.
1194 N. Mathilda Avenue

Sunnyvale, CA 94089
United States

Date Issued:	05 AUG 2005	Customer PO:	
Order Number:		Part Number:	
Quote Line:	1	Part Description:	

Juniper RTU (Right to Use) Certificate

Instructions:

New Juniper products typically ship with a temporary unlimited license that expires after a period of time if a permanent license key is not installed (see product literature for exact expiration period).

To obtain a permanent license key for new products or upgrade license keys for already deployed products, please visit https://www.juniper.net/generate_license to convert the below Authorization Codes into license keys.

Once you have obtained the license keys they can be loaded onto your hardware to unlock your purchased features.

If you have any questions about the license generation process, please contact your reseller or distributor.

Item#	RTU Serial Number	Authorization Code
1	RTU00000	-unin-
2	RTU00000	-unin-

Software upgrades retain the Permanent License keys (see Upgrading the DX Application Acceleration Platform on page 68).

Obtaining a Juniper Customer Support Center (CSC) User ID and Password

Before you can obtain a permanent license, you must have a Juniper Customer Support Center (CSC) User ID and Password. There are two ways of obtaining these:

- Call Juniper Customer Care at 1-800-638-8296 (United States) or +1-408-936-1572 (outside the United States).
- You Can Register online at the Customer Support Center:

  https://www.juniper.net/generate_license

Obtaining a Permanent License

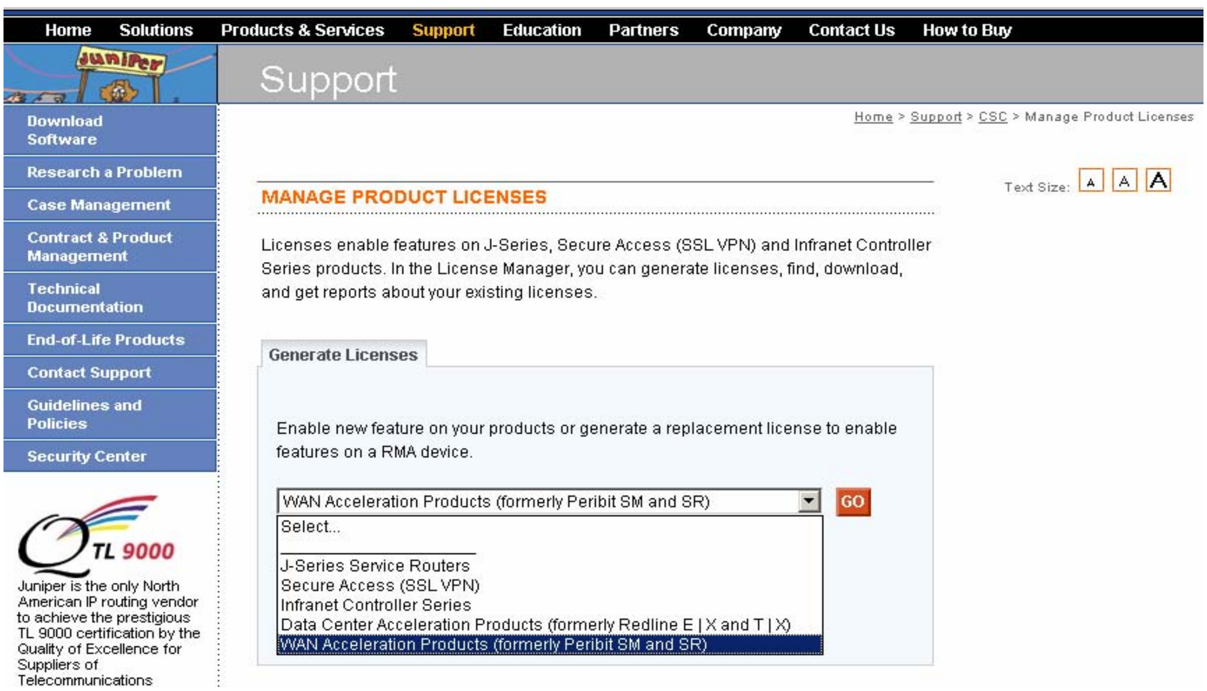
To obtain a permanent license, from a standard web browser, go to the web site:

https://www.juniper.net/generate_license

and log in using your assigned User ID and Password.

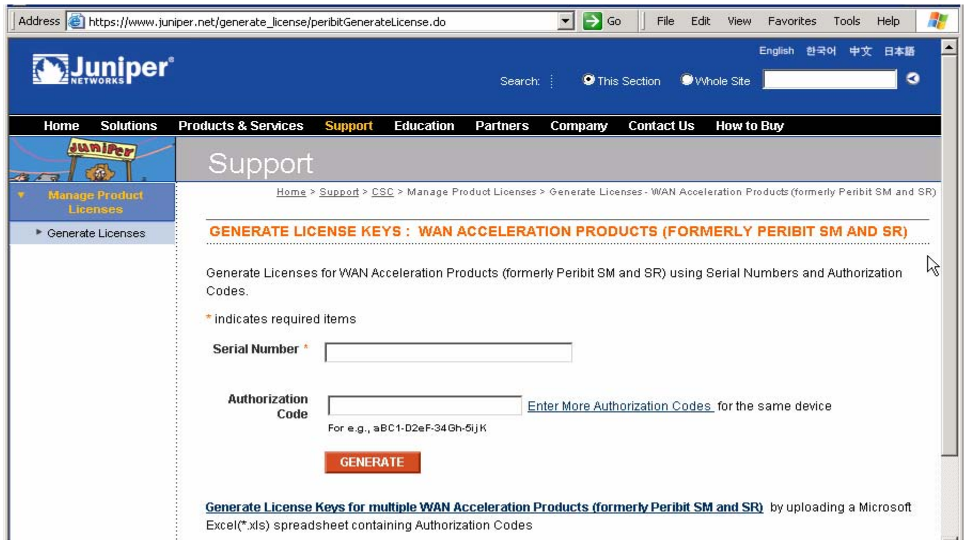
Select the "WAN Acceleration Products (formerly Peribit SM and SR)" link (Figure 15) and click on the **Go** button..

Figure 15: Manage Product Licenses Screen



This will bring you to the Generate License Key screen (Figure 16).

Figure 16: Generate License Key Screen



Enter the serial number for your DX Application Acceleration Platform, and then enter your Authorization code. Click on the **Generate** button to continue.

The next screen will display the license key. to save it, copy the key from the browser screen and paste it into a text document.

Installing the DX License Key

If you are upgrading from a version of the software earlier than 3.1, you will need to install the DX appliance license key. If you are upgrading from version 3.1, the previously installed license key is preserved, and you do not need to reinstall it.

1. Obtain the license key specific to the serial number of your DX Application Acceleration Platform as detailed in Obtaining a Permanent License on page 48. Each license key will work ONLY on the specified DX appliance.

2. Install the DX appliance license key by copying and pasting the license key into the console using the following command:

```
dx% capture license
```

A sample of the license key is:

```
----------BEGIN JUNIPER LICENSE KEY----------
3418fa1db0e0ae0552cb79c472a076b15a9a4b51d0e2f6a54fcfa97e4b04b20f
29f4149978330387d102e076805e884bcf0f14d023db999b79651e140c732431
f3b3b815d3cf3eb593060e917c458defe8267da03a3ee3101d99c9becd34643d
6184fc028ff719bcd451f87ad431f90c28d6c68e85d105443edfbfe772d7df8b
426f3cd08ba32863c37ba856139af4169d7102f53aabbc3f688a8b171c3446e9
f0819b23dd4f0bea49c9ae3b1ecb9feef5361ca3a9
----------END JUNIPER LICENSE KEY----------
```

Administrator Audit Trail

Overview

The Audit Trail provides a log of all activities performed on the DX Application Acceleration Platform . Specifically, it provides the following information:

- Timestamp
- Location of the source
- Username
- Access method: Web User Interface (WebUI) vs. DXSHELL
- Changes and activities performed

Entries are only added to the audit log after a `write` operation has been performed.

Syntax of the Log Entries

The following is the syntax of the audit trail:

```
<Timestamp> <Location> <Username> <Tool> <Change>
```

where:

- `<Timestamp>`: The time of the change, displayed in [YYYY-MM-DD HH:MM:SS (TZ Offset)] format. For example: `[2004-10-04 22:17:09 (+0700)].`
- `<Location>`: The information of the system that made the change. It can either be the IP address of the workstation, or the reserved word "console" if the change is made from the serial console.
- `<Username>`: The username of the user that made the change. When changes are made by a module internal to the DX appliance, the username listed is "RLN Internal".
- `<Tool>`: Indicates if the change was from DXSHELL, the Web User Interface (WebUI), or an internal module. The word "DXSHELL" indicates that the change was made from the command line interface using the serial console, SSH or Telnet. The word "WebUI" indicates that the change was made from the WebUI. The word "system" indicates that the change was made by a DX appliance.
- `<Change>`: Indicates what was changed.

To display the audit trail, use the following command:

```
dx% show log audit
```

Enabling and Disabling Logging of "show" Commands

The logging of `show` commands is disabled by default. It can be enabled and disabled using DXSHELL or the Web User Interface (WebUI).

To enable logging of `show` commands, use the command:

```
dx% set admin audit showcmd enabled
```

To disable logging of `show` commands, use the command:

```
dx% set admin audit showcmd disabled
```

All commands executed on DXSHELL are logged when a `write` operation is performed.

To display the audit trail settings, type the command:

```
dx% show admin audit showcmd
```

Event Logging and Notification

By default, the DX appliance keeps a log of a number of system events. You can view the DX appliance's event log by typing the command:

```
dx% show log system
```

You can view the configuration of event logging by typing the command:

```
dx% show admin log
```

To keep an eye on system performance and status, you can also configure the DX appliance to report certain classes of events using E-Mail or logging them to an external syslog logging machine.

There are four ways to record DX appliance system events:

- console: Shows log events on the system console
- e-mail: Sends notifications of log events to specified E-Mail addresses
- syslog: Sends log events to an external syslog logging server
- memory: Records log events on the DX appliance

There are two levels of events; from most serious to least, they are:

- EMERG
- ALERT

If you set your alert level to ALERT, you will get both EMERG and ALERT notices. If you set your alert level to EMERG, you will only get EMERG notices. For a complete list of events for each level, refer to "List of Events" on page 371.

Example: Receive Notification of Layer 7 Health Check Errors using E-Mail

1. Make sure event logging is enabled by typing:

   dx% **set admin log enabled**

2. If you have not already specified an SMTP server that the DX appliance can use to relay E-mail, specify one by typing:

dx% **set admin email server <IP address or hostname of SMTP server>**

3. Specify the from E-Mail address that should appear in the E-Mail by typing:

   dx% **set admin email from <e-mail address>**

4. Specify up to two E-Mail addresses that should receive event notifications by typing:

   dx% **set admin log mailto1 <e-mail address>**
   dx% **set admin log mailto2 <e-mail address>**

5. Set the threshold for E-Mail event notification low enough to include L 7 health check errors. L 7 health check errors are [ALERT] level events, so you enable notification by typing the command:

   dx% **set admin log email ALERT**

   to ensure that E-Mail notification is sent for all ALERT level events or greater.

Configuration Management

You can export and import an DX appliance's configuration:

- For backup purposes
- To simplify matching settings across multiple units

Exporting and importing requires:

- Access to the DXSHELL command line
- A TFTP or SCP server

Exporting a Configuration

To export a configuration, enter the command:

```
dx% export config tftp://<tftpservername>/<configfilename>
```

or

```
dx% export config scp://<scpservername>/<configfilename>
```

The `scpservername` is a host name or an IP address. The filename is an absolute path for the file where you would like to export the configuration. The directory specified for the filename must exist.

Note when exporting a configuration, it is a good practice to give the file a name that describes the function and identifies the version.

For example:

```
dx_4.1.B14_ssl_server_8-25-2003.
```

Note that the following settings are NOT exported:

- Passwords
- Commands that control the DX appliance server's state
- Commands that control the state of administrative services (e.g., SSH, Telnet and WebUI access)
- Ssl Keys And Certificates
- User Account Information
- Application Rules
- License Keys

View the Contents of a Configuration File

To view the contents of the DX appliance's current configuration file, use the command:

```
dx% display config
```

Importing a Configuration

To import a configuration, use the command:

```
dx% import config tftp://<tftpservername>/<configfilename>
```

or

```
dx% import config scp://<scpservername>/<configfilename>
```

If any errors are encountered in the configuration file, the DX appliance will generate an error message and stop the import process on the error.

Apply and save the new configuration with the command:

```
dx% write
```

NOTE: SSL keys and certificates are not included in an exported configuration for security reasons. When importing a configuration, you must make sure that the required SSL keys and certificates are already installed on the DX appliance.

Editing a Configuration

Configuration files can help you match common settings across multiple DX appliances. A freshly exported configuration file contains settings for things like IP addresses that are particular to a single DX appliance.

You can edit the configuration file, removing distinct settings, to create a general configuration file that your DX appliances can share. You can also create a partial configuration file to match a particular subset of settings across multiple DX appliances.

Configuration File Formats

An exported DX appliance configuration file consists of the list of DXSHELL commands required to completely recreate a configuration. Upon importing a configuration file, the DX appliance will run the commands in the file to reproduce the configuration.

You can use a text editor to customize a configuration file, removing or commenting out commands, so that the file contains only settings which can be shared by multiple DX appliances.

Lines beginning with a "#" are ignored:

```
# This is a comment
```

NOTE: A freshly exported configuration file begins with the command:

```
dx% copy config factory memory
```

This command resets all settings to factory defaults before applying the commands in the configuration file. To avoid losing your configuration, DO NOT include this command in custom partial configuration files.

Example: Partial Configuration for Sticky Load Balancing

For example, if you wanted to match sticky load balancing settings across several DX appliances you could create a partial `config` file with only sticky load balancing settings, and then import this `config` file onto each DX appliance.

In the example below, the TFTP server's address is 192.168.0.11 and the name of the `config` file is `juniper_sticky.conf`.

1. Export the configuration file from the DX appliance with the correct sticky load balancing settings by typing:

   ```
   dx% export config tftp://192.168.0.11/juniper_sticky.conf
   ```

2. Open the file in a text editor and remove all commands not related to sticky load balancing and leave just the following commands in the file:

   ```
   set cluster 1 sticky clientip distribution internet
   set cluster 1 sticky clientip timeout 120
   set cluster 1 sticky cookie expire 0
   set cluster 1 sticky cookie mask ipport
   set cluster 1 sticky method none
   ```

   For partial configuration files, be sure to remove the command:

   ```
   dx% copy config factory memory
   ```

   which resets all settings to their factory defaults. Save your changes.

3. On each of the DX appliances where you wish to share the sticky load balancing settings, enter the command:

   ```
   dx% import config tftp://192.168.0.11/juniper_sticky.conf
   ```

Restoring the Factory Default Configuration

To erase all custom settings and return to the factory default configuration, use the command:

```
dx% reset config
```

CAUTION: If you are connected to a DX appliance remotely, you must enter valid network settings BEFORE applying the new configuration or you will no longer be able to connect to the DX appliance. If you leave the factory network settings in place you will have to connect to the DX appliance's console port to enter new network settings.

To save and apply the factory default configuration, enter the command:

```
dx% write
```

System Snapshot

System Snapshot creates an image of the system, including IP addresses, system files, and licenses. System Snapshot creates an effective backup of not only the configuration, but also the underlying operating system. Some organizations find it convenient to have base system images that can be used to clone a new machine at a moment's notice.

System Snapshot allows these advantages:

- Repair by replacement allows more uptime: Field units that have failed can be quickly recreated using replacement hardware units and a system snapshot of the former unit.
- Recovery from configuration mistakes: Administrators can revert to a known system snapshot should they ever want or need to return to a previous configuration.
- All of the information that is exported during a system snapshot is encrypted. It is not casually visible if viewed off the DX appliance.

After a system snapshot import and subsequent system reboot, all services that were running on the original machine when the snapshot was taken will start on the machine that imported the system snapshot. System snapshot is imported into an unused partition. Importing a snapshot does not impact the running (active) partition.

Each DX appliance has a manufacturing information file that contains the unit serial number, manufacturing date, model number and platform at the time of manufacturing. This information is not overwritten when importing a system snapshot.

A system snapshot requires:

- Access to the DXSHELL command line with the role of Administrator
- A SCP server

A system snapshot is invoked by typing the command:

```
dx% export snapshot system scp://<server>/<path>/<resource>
```

For example:

```
dx% export snapshot system scp://myarchive/usr/cvs/snap_juniper1
Creating...
myuser@myarchive's password:
Successfully exported snapshot.
```

A previously saved system snapshot is restored by typing the command:

```
dx% import snapshot system scp://<server>/<path>/<resource>
```

For example:

```
dx% import snapshot system scp://myarchive/usr/cvs/snap_juniper1
```

This system shows messages similar to this:

```
WARNING - This will import a snapshot and install it to the alternate
partition, overwriting its current contents.

After the import, you will be able to choose whether to use the snapshot or
currently active settings for your license and network configuration. You
can use the 'set boot' command to select the default boot partition once
this process is complete.

Would you like to continue (y/n)? [y]

Receiving file /usr/cvs/snap_juniper1 from scp server myarchive...
myuser@myarchive's password:
```

If you answer “no” the system replies with:

```
Import aborted.
```

If you answer “yes” the system replies with the normal “set server down” message and status message:

```
myuser@myarchive's password:
Bytes received: 13054697
Verifying...
Decrypting...
Unpacking...
Installing............................
Verifying install...........................................
Doing post-install setup...
Done.
                        From snapshot:          Currently active license:
License:                INVALID                 Valid

The license from the snapshot is used by default. Would you like to use the
currently active license instead (y/n)? [n]

[Installing currently active license to | Using snapshot license on]
alternate partition.

                        From snapshot:          Currently active settings:
Hostname:               dx-1.domain.com         dx-2.domain.com
Default route:          192.168.0.1             10.0.51.1

Ether 0 IP Address:     192.168.14.20           10.0.51.80
Ether 0 Netmask:        255.255.0.0             255.255.255.0
Ether 0 Media:          100baseTX full-duplex   autoselect
Ether 0 MTU:            1500                    1500

Ether 1 IP Address:     192.168.14.21           10.0.51.81
Ether 1 Netmask:        255.255.0.0             255.255.255.0
Ether 1 Media:          autoselect              autoselect
Ether 1 MTU:            1500                    1500
```

If there are more Ethernet cards on the snapshot than in the current machine, you see a dialog similar to this:

```
Ether 2 IP Address:     192.168.14.20          Ether 2 not present
Ether 2 Netmask:        255.255.0.0            n/a
Ether 2 Media:          100baseTX full-duplex  n/a
Ether 2 MTU:            1500                   n/a
```

If there are more Ethernet cards in the current machine than on the snapshot you see a dialog similar to this:

```
Ether 2 IP Address:     Ether 2 not present    10.0.51.82
Ether 2 Netmask:        n/a                    255.255.255.0
Ether 2 Media:          n/a                    100baseTX full-duplex
Ether 2 MTU:            n/a                    1500
The network settings from the snapshot are used by default. Would you
like to use the currently active settings instead (y/n) ? [n]
[Installing currently active network settings to | Using snapshot network
settings on] alternate partition.
Import snapshot successful. Use 'set boot' to activate the new
partition, and 'reboot' to switch to it.
dx%
```

In most cases, the server can continue to run when system snapshots are exported and imported However, under certain conditions caused by memory constraints, the server may have to be stopped before exporting or importing the snapshot. In these cases, the user will be prompted to stop the server before export or import, and to restart the server afterward.

Here is an example of an export:

```
dx% export snapshot system scp://myarchive/usr/cvs/snap_juniper1
WARNING -   Because of memory constraints, the server will be stopped
            if the export continues.
Would you like to continue (y/n)?[y]
Running 'set server down'...
The EIX server was stopped.

Creating...
myuser@myarchivels password:
Successfully exported snapshot.
The server is currently down.
Would you like to start it now (y/n)?[y]
Running 'set server up'... The EIX server was started.
dx%
```

Configuration Synchronization

Configuration synchronization provides a mechanism for an administrator to copy the settings from one DX appliance to other DX appliances in a pre-defined group. This can be a significant time-saver for administrators with two or more DX appliances.

Examples of information that is synchronized across the group are:

- SSL Certificates, Keys, and Passwords
- OverDrive Rules
- Username and Password Combinations

However, there are settings that must be unique to a machine within in the group. These are called exceptions. The exceptions currently include:

- Hostname
- Ethernet IP Addresses, Netmask, and other Interface Settings
- Default Route
- Administration VIP and Administration Interface IP Addresses
- Administration SOAP Settings

A synchronization override file is used to manage values for these synchronization exceptions. It contains a list of synchronization addendums that contain user-entered information that will be synchronized across machines in the group. This synchronization override file can be used during future configuration synchronizations.

NOTE: Use of the synchronization override file is important. If Configuration Synchronization is performed in a production environment without using the synchronization override file to manage the synchronization exceptions, unexpected network behavior can result.

Synchronization is achieved through the use of a SOAP server. Simple Object Access Protocol (SOAP) is a XML based protocol for exchanging information over the Internet. Commands are provided to manage both the Synchronization Group and the SOAP Server. In order for Configuration Synchronization to work, SOAP must be configured and enabled on the group member unit(s). If SOAP is not configured and enabled, then Configuration Synchronization will fail. SOAP does not have to be configured and enabled on the reference unit, but will cause no problems if it is configured.

NOTE: Configuration Synchronization is **NOT** supported on the DX 3650-FIPS version of the DX Application Acceleration Platform .

Configuration Synchronization Override File Format

The synchronization override file is a text file, and while its entries are fully managed by the configuration synchronization feature, it also can be edited manually. This is an example of a synchronization override file:

```
# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE

# Description: Sample manual override command file for
#              configuration synchronization.
#
# Note: The first line of this file must be the text inside the double
quotes:
#       "# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE"
#
# Example:
#   1) sync group consists of two member appliances: dx1 and dx2
#   2) cluster "1" exists in the configuration
#
# Here are some typical commands that might be unique for each member:
#
dx1 | set cluster 1 listen vip 192.168.0.10
dx1 | set activeN failover nodeid 1
dx1 | set slb failover nodeid 1

dx2 | set cluster 1 listen vip 192.168.0.20
dx2 | set activeN failover nodeid 2
dx2 | set slb failover nodeid 2
```

Note that the first line of this file must be the text:

```
# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE
```

Configuration Synchronization Commands

To synchronize the configuration settings across a group of DX appliances, type the command:

```
dx% sync group <name>
```

Before this command can be executed, both the Synchronization Group and the SOAP server must have been set up correctly. When adding members to the synchronization group, the local DX appliance must be added as a reference unit along with all remote DX appliances that need to receive the group configuration. For example:

```
dx% add sync group test
Created sync group "test"
dx% add sync group test member 10.0.10.100
Created sync group "test" member "10.0.10.100".
dx% add sync group test member dx-1
Created sync group "test" member "dx-1".
dx% set sync group test member
10.0.10.100  dx-1
dx% set sync group test member 10.0.71.100 password
New password:
Retype new password:
dx% add sync group test member dx-1
Created sync group "test" member "dx-1".
dx% set sync group test member dx-1 password
New password:
Retype new password:

dx% show sync group
Sync Group [test]
Description:
Override Filename:
Override Status: disabled
Timeout: 180
Member:                     Port:         Username:         Password:
  10.0.10.100               8070          admin             ********
  dx-1                      8070          admin             ********

dx% sync group test
Using the current appliance as the reference, you are about to
synchronize the configuration settings on the following appliances:
    10.0.10.100 (reference)
    dx-1

All settings except the following will be synchronized:
    Hostname
    All ether settings
    Default route
    Admin VIP and interface settings
    Admin soap settings

You have specified the following manual override file:
    dx70_override (enabled)
Would you like to continue (y/n)? [y]
Synchronizing member 10.0.10.100 ...
Success (skipping sync with localhost).
Synchronizing member dx-1 ...
Success.
Synchronization for group "test" finished successfully.
```

Synchronization Group Management Commands

For each of the commands, `<memberid>` is either a `<hostname>` or an `<ip>`.

To create a synchronization group, type the command:

```
dx% add sync group <name>
```

To add a member to the synchronization group, type the command:

```
dx% add sync group <name> member <memberid>
```

To set the username for a synchronization group member, type the command:

```
dx% set sync group <name> member <memberid> username <username>
```

The default username is "`admin`".

To set the password for a synchronization group member, type the command:

```
dx% set sync group <name> member <memberid> password
```

This will prompt you for a password. For example:

```
dx% set sync group <name> member <memberid> password
New password:
Retype new password:
dx%
```

No asterisks will be shown as the password is typed.

To add a description for a synchronization group, type the command:

```
dx% set sync group <name> description <description>
```

To rename a synchronization group, type the command:

```
dx% set sync group <name> name <newname>
```

To enable the use of the group override file, type the command:

```
dx% set sync group <name> override enabled
```

To disable the use of the group override file, type the command:

```
dx% set sync group <name> override disabled
```

The default is disabled.

To set the name for the group override file, type the command:

```
dx% set sync group <name> override filename <filename>
```

To show all of the settings for the synchronization group, type the command:

```
dx% show sync group
```

To show all of the settings for a particular synchronization group, type the command:

```
dx% show sync group <name>
```

To show the settings for a particular synchronization group member, type the command:

```
dx% show sync group <name> member
```

To show the description for a particular synchronization group member, type the command:

```
dx% show sync group <name> description
```

To show the override status for a particular synchronization group, type the command:

```
dx% show sync group <name> override
```

To show the override filename for a particular synchronization group, type the command:

```
dx% show sync group <name> override filename
```

To delete a synchronization group, type the command:

```
dx% delete sync group <name>
```

To delete a member from a synchronization group, type the command:

```
dx% delete sync group <name> member <memberid>
```

SOAP Server Management Commands

To enable the Simple Object Access Protocol (SOAP) server, type the command:

```
dx% set admin soap up
```

The default is up.

To disable the SOAP server, type the command:

```
dx% set admin soap down
```

To set the port number for the SOAP server, type the command:

```
dx% set admin soap port <portnum>
```

The default port is 8070.

To set the SSL certfile filename for the SOAP server, type the command:

```
dx% set admin soap ssl certfile <filename>
```

The default file name is democert.

To set the SSL key file for the SOAP server, type the command:

```
dx% set admin soap ssl keyfile <filename>
```

The default file name is demokey.

To set the SSL key password for the SOAP server, type the command:

```
dx% set admin soap ssl keypass <password>
```

To show all of the configuration parameters for the SOAP server, type the command:

```
dx% show admin soap
```

To show the port number for the SOAP server, type the command:

```
dx% show admin soap port
```

To show all of the SSL configuration parameters for the SOAP server, type the command:

```
dx% show admin soap ssl
```

To show the SSL certfile filename for the SOAP server, type the command:

```
dx% show admin soap ssl certfile
```

To show the SSL key file filename for the SOAP server, type the command:

```
dx% show admin soap ssl keyfile
```

To show the SSL key file password for the SOAP server, type the command:

```
dx% show admin soap ssl keypass
```

Configuring the Login Banner

Some users have corporate policies that require them to display a login banner to users when they are accessing corporate computer systems and networks. You can customize the welcome message that is displayed on either the Juniper Command Line or on the WebUI when a user logs in. Currently, when a user logs in using DXSHELL, the default message is displayed:

```
        Welcome to Juniper Networks
                    DX
      Application Acceleration Platform
```

The login banner could be changed to include any message that you would like to display. For example:

```
Unauthorized access to or use of this system is prohibited. All access
and use may be monitored and recorded.
```

The maximum length of the text string is 2000 characters. The banner allows for some printf-style substitutions, as follows:

```
%h hostname
%d date
%s system ("Juniper")
%v product version
%b product build id
%% show the percent character
```

When the banner display encounters one of these substitution strings, it extracts the information from the appropriate place in the operating system and displays it. This information cannot be changed by the user.

The banner cannot be exported. The banner is preserved when installing newer versions of software for the DX appliance.

Configuring the Login Banner from the Command Line Interface

The login banner can only be configured from the Command Line Interface. These commands are available to configure the banner:

```
dx% capture loginbanner
```

This command begins capture of the login banner. After typing this line has been entered, everything that you type (up to the 2000 character limit) will be captured as part of the banner to be displayed. Carriage returns can be included, and you can use copy and paste commands to make the capture process easier. End capture of the banner by typing a period on a blank line. This command can only be executed by a user with a role of “administrator.”

```
dx% display loginbanner
```

This command displays the banner in its raw form. Substitution strings are shown in their normal form (%h) instead of the substitution form (hostname). The `display loginbanner` command can only be executed by a user with a role of “administrator.”

```
dx% delete loginbanner
```

The `delete loginbanner` command deletes the banner. It can only be executed by a user with a role of "administrator."

```
dx% show loginbanner
```

The `show loginbanner` command shows the banner with the appropriate substitutions. It is available to all users.

Capturing a Login Banner

Follow these steps to capture a login banner:

1. Type the `capture loginbanner` command:

   ```
   dx% capture loginbanner
   ```

2. Enter the information that you want to display. End with a period on a blank line.

   ```
   Unauthorized access to or use of this system is prohibited.
   All access and use may be monitored and recorded.
   %h
   %d
   %s
   Put anything else that you want here . . .
   .
   ```

3. Type the `show loginbanner` command to show the banner with the appropriate substitutions:

   ```
   Unauthorized access to or use of this system is prohibited.
   All access and use may be monitored and recorded.
   MyFirstHost
   4 July 2004
   Juniper
   Put anything else that you want here . . .
   ```

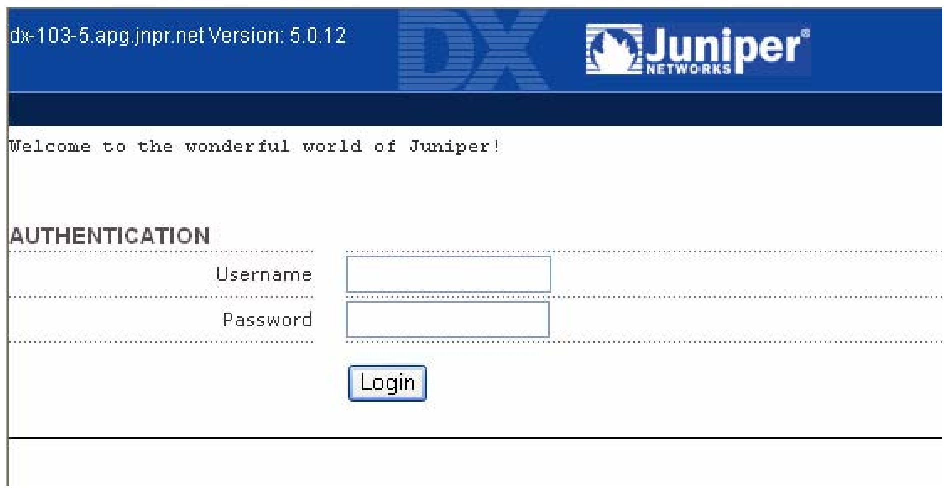
Displaying the Login Banner in the Web User Interface

The WebUI does not provide the capability to set the login banner. Instead, the text string that was set by the administrator using the `capture loginbanner` command is displayed as part of the WebUI login screen.

For example, if the administrator sets the login banner “Welcome to the World of Juniper Networks” using the command string:

```
dx% capture loginbanner
Welcome to the wonderful world of Juniper.
.
```

A new instance of the WebUI will display the following:



NOTE: You can put HTML in your login banner, and it will display correctly on the WebUI. However, the DX appliance does not parse out HTML code when displaying the banner on the Command Line Interface, so the HTML code will be displayed along with the desired banner.

Upgrading the DX Application Acceleration Platform

WARNING: When upgrading, the DX appliance should not be handling live traffic, as the upgrade will interrupt the traffic flow and requires a reboot.

DX Application Acceleration Platform License Key

A DX appliance unit must have a license key installed to enable software features running on the system. The license key provides information about the hardware and the software features that the system is licensed to run. To see the features that are currently enabled in your system, type the command:

```
dx% show license
```

If you do not have a license key, or the license key is missing for the DX appliance, follow the instructions in The License Key on page 47.

Upgrade Requirements

To upgrade an DX appliance unit, you need:

- A TFTP or SCP server to hold the upgrade file
- An upgrade file (.pac) that corresponds to your DX appliance model

The procedure for configuring a TFTP or SCP server varies from system to system. If you need to set up a TFTP or SCP server, consult the documentation for your operating system.

Upgrade files can be obtained from your Juniper Networks sales representative or reseller, or from the Juniper Networks Technical Support web site.

Preserve Your Configuration and Choose a .pac File

1. Put the upgrade file on your TFTP or SCP server. By default, most unix-based TFTP servers expect files to be located in `/tftpboot`.

2. Preserve the DX appliances configuration.

   The steps required to preserve your configuration depends upon which software release your DX appliance is running. To determine the version number of the software release your DX appliance is running, use the command:

   ```
   dx% show version
   ```

For Version 2.1 and Greater

If your DX appliance is running release 2.1 or later, your configuration will automatically be preserved when you upgrade.

For Versions Prior to 2.1

If you are upgrading from a version prior to 2.1, you will have to re-configure the DX appliance after upgrading, so be sure to write down your configuration before upgrading. To view a complete summary of your configuration, use the command:

```
dx% show config
```

3. Configure the DX appliance to use your TFTP or SCP server.

   a. Give the DX appliance the IP address of your TFTP or SCP server:

   ```
   dx% set admin tftp server <IP address of TFTP server>
   ```

   b. Tell the DX appliance which file to retrieve from the TFTP server:

   ```
   dx% set admin upgrade filename <name of upgrade .pac file>
   ```

4. Save the changes:

   ```
   dx% write
   ```

5. Verify the TFTP configuration:

   ```
   dx% show admin upgrade filename
   dx% show admin tftp
   ```

Upgrading Using the `install` Command

The `install` command preserves the current version of the firmware. With the `install` command, you keep the current working version on the active partition, while installing the newer version into the non-active partition. This allows you to test the new firmware and easily revert to the previous version if needed. Note that the `.pac` file for the `install` command is approximately 14 MBytes.

You must be logged in as a user with a role of *administrator* to perform the install procedure.

1. Ensure that the DX appliance is not actively handling traffic. by typing the command:

   ```
   dx% set server down
   ```

2. View the setup to see the partition where the new firmware will be installed by typing:

   ```
   dx% show boot
   ```

   A sample output is shown as follows:

```
Boot 1 (cur, act) : Juniper Networks Accelerator DX 2.3.3 Wed Mar 12 20:35:50 GMT
Boot 2          : Empty partition
```

   The current partition (cur) is the partition that is currently running. The active partition (act) is the one that will be used after the reboot. In this example, the current partition and the active partition are the same.

3. Copy the `install pac` file to a TFTP or SCP server that is accessible by the machine being upgraded.

4. Configure the TFTP or SCP server and the name of the install file on the DX appliance by typing:

```
dx% set admin tftp server <IP address or Hostname of the tftp server>
```

```
dx% set admin upgrade filename <install_filename.pac>
dx% write
```

Where `install_filename.pac` is the filename of the install file on the TFTP server.

5. Store a copy of the existing configuration for backup purposes by typing:

```
dx% export config tftp://<tftp_server>/dx_configuration
```

Where `tftp_server` is the IP address of the TFTP server and `dx_configuration` is the filename of the saved configuration on the TFTP server.

6. Install the new **firmware by typing:**

```
dx% install
```

This will download the .pac file from the TFTP or SCP server specified.

7. You will be prompted with the following warning. Enter **y** to continue.

```
WARNING - this will install firmware to the alternate partition, and will
overwrite its current contents. After the install, you will be able to
select the default boot partition. You will also be given the option to copy
your existing configuration to the new partition.

Would you like to continue (y/n)? [n]
```

8. After the install completes, set the boot partition for the next reboot by typing:

```
dx% set boot 2
```

Typing the `show boot` command will now show something like the output below. The active partition is the one where the DX appliance will boot.

```
Boot 1 (current) : Juniper Networks Accelerator DX 2.3.3 Wed Mar 12 20:35:50
GMT
Boot 2 (active)  : Juniper Networks Accelerator DX 3.1.0 Fri June 9 20:35:56
GMT
```

9. Reboot the DX appliance:

```
dx% reboot
```

10. Log into the DX appliance using the default username `admin` and the password that you previously defined as the default password, before you installed the new release. If you did not change the default password, it is set to `juniper` by the factory.

11. Import your old configurations and restarting of services. You will be prompted to import your previous configuration settings with the following question:

```
"Would you like to import your existing configuration? [y|n]"
```

Select "yes" to import your previous configuration into the new install. Select "no" if you are not interested in importing your previous configuration at this time.

If you selected “yes”, you will next be asked:

“Would you like to save your current configuration? [y|n]”

Select “yes” to save the configuration to the disk. Select “no” if the configuration shown is not the one you wanted.

Next you will be asked:

“Would you like to restart your services? [y|n]”

Select “yes” if you would like the services running in your previous configuration to be restarted in the new installation. Select “no” if you do not want any services started at this time. You can enable services later as an administrative user.

12. Save imported settings with the write command:

    dx% **write**

13. Verify that the software was upgraded to the intended version by typing the command:

    dx% **show version**

14. Make sure that the configuration is correct by typing the command:

    dx% **show config**

15. Optionally, you may set up one or more users to administer the DX appliance. Refer to the procedures in “Multi-Level Administration Rights” on page 35.

    For security reasons, the SSL keypass (pass phrase) is not copied over as part of the configuration file on the new partition after an upgrade. You can import the keypass by typing command:

    dx% **set cluster <n> listen ssl keypass <key password>**

NOTE: During an install, the configuration files are copied to the non-active partition. If you reboot to the alternate partition immediately, then the most recent configuration files are used, and no problems should be encountered.

However, you may choose to install the software, and then reboot the DX appliance at a later time to limit network impact. If changes are made to the configuration between the install time and the reboot time, the configuration files on the alternate boot partition are no longer current. You must then do another install just before rebooting the unit in order to have updated configuration files on the alternate boot partition.

Chapter 6

Integrating the DX Appliance into Your Network

This chapter describes how to integrate your DX Application Acceleration Platform into your network, discussing the following topics:

Overview

The last step in fully-integrating the DX Application Acceleration Platform into your network is to direct incoming web requests to the DX appliance rather than to your web server.

Sample Network Topologies

The DX Application Acceleration Platform is designed to work with any network architecture. The diagrams on the following pages provide examples of some of the ways in which you can deploy one or multiple DX appliances to accelerate:

- A Web Cluster
- A Web Farm (multiple web clusters)
- Reverse proxy cache
- Three-tier Enterprise Applications (e.g., CRM applications)
- Remote Access
- Sites that use cookie-based load-balancing (refer to “Setting up the DX Appliance for “Sticky” Traffic” on page 155 for additional information)
- Sites that use SSL (refer to “Setting Up the DX Appliance for SSL Traffic” on page 159 for additional information)

Web Cluster

Figure 17: Accelerating a Web Server Cluster with the DX Appliance (In-Line)

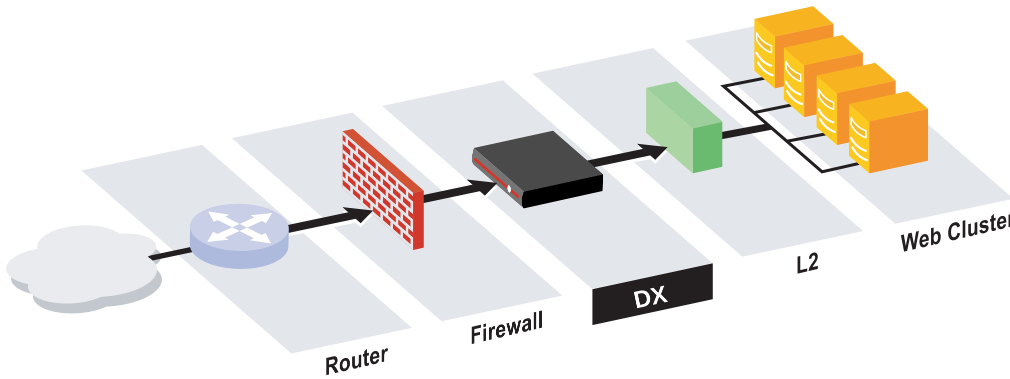
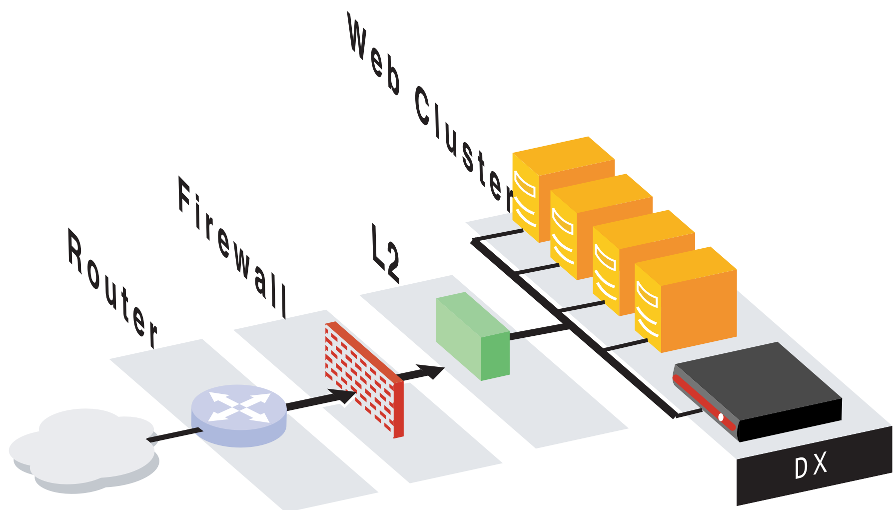


Figure 18: Accelerating a Web Server Cluster with the DX Appliance (One-Arm)



Web Farm

Figure 19: Accelerating a Web Farm with the DX Appliance (In-Line)

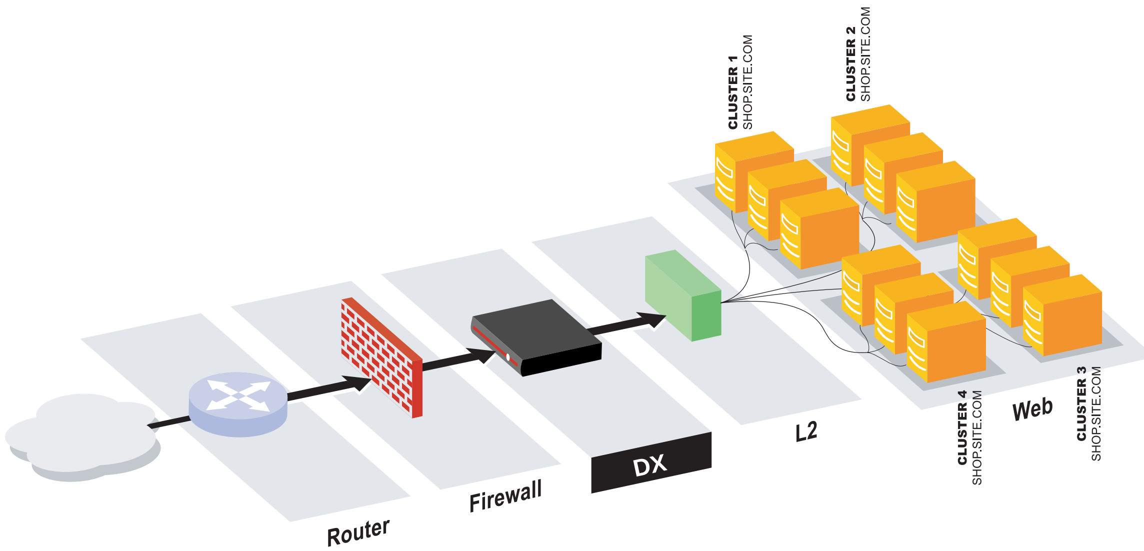
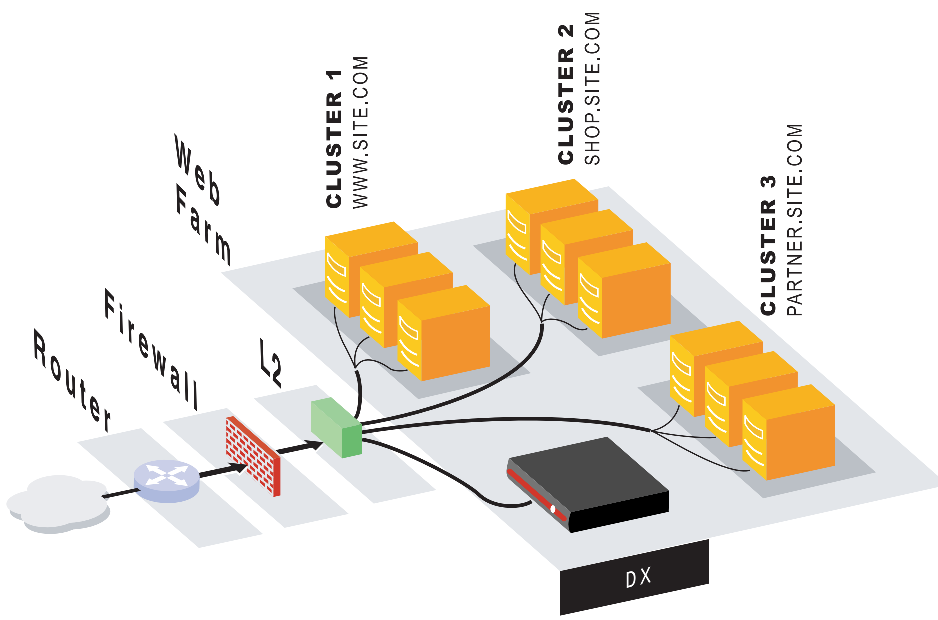


Figure 20: Accelerating a Web Farm with the DX Appliance (One-Arm)



Reverse Proxy Cache

Figure 21: Accelerating Reverse Proxy Cache with the DX Appliance (In-Line)

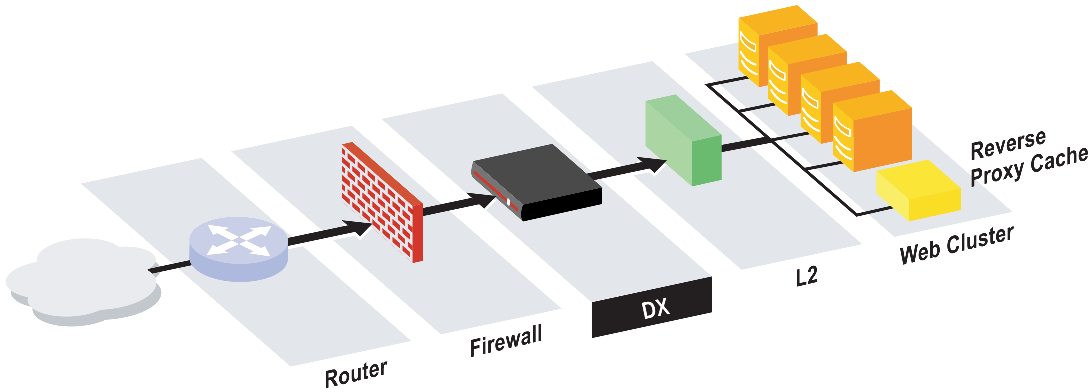
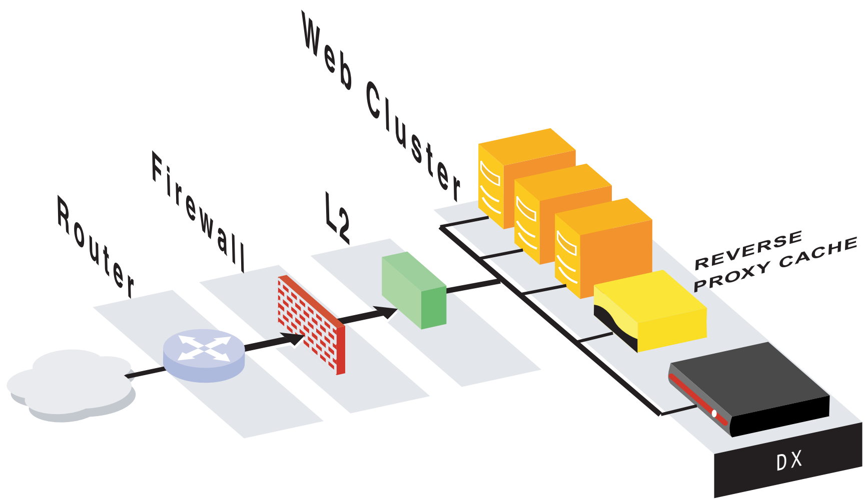


Figure 22: Accelerating Reverse Proxy Cache with the DX Appliance (One-Arm)



Three-Tier Enterprise Application

Figure 23: Accelerating a Three-Tier Enterprise Application with the DX Appliance (In-Line, e.g., CRM Applications)

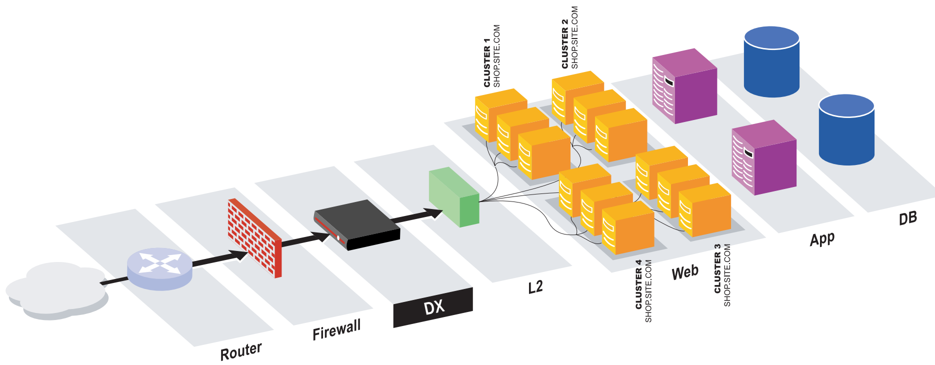
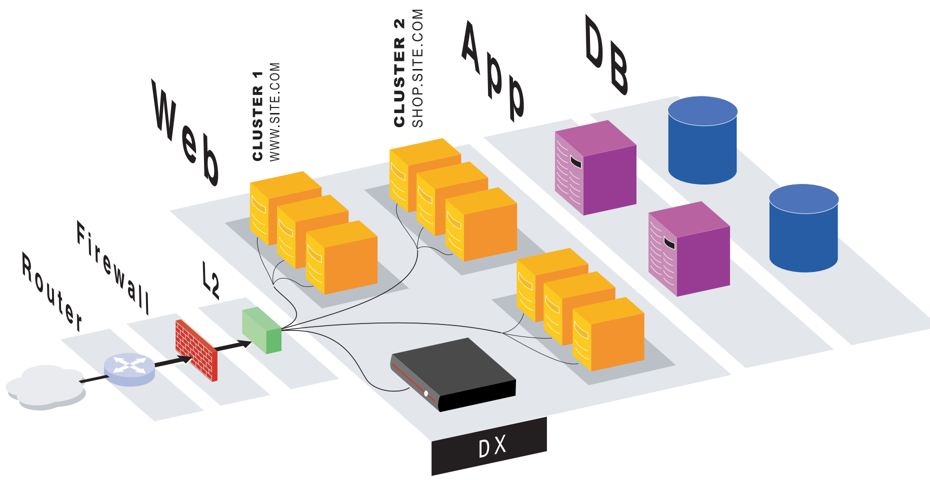


Figure 24: Accelerating a Three-Tier Enterprise Application with the DX Appliance (One-Arm, e.g., CRM Applications)



Remote Access

Figure 25: Accelerating Remote Access to Corporate Network and Web Applications (In-Line)

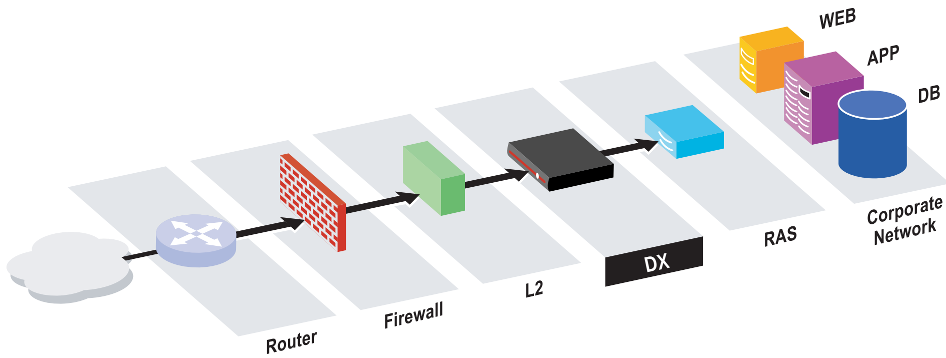
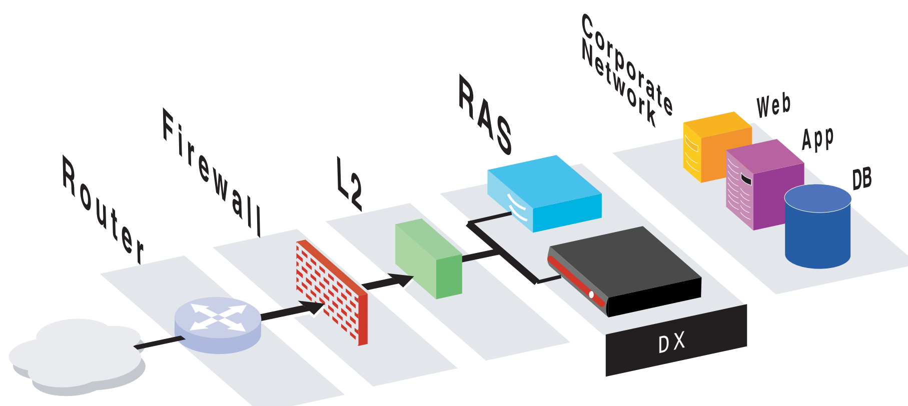


Figure 26: Accelerating Remote Access to Corporate Network and Web Applications (One-Arm)



Deploying the DX Appliance Behind an External Server Load Balancer (SLB)

NOTE: The DX appliance has a built-in Server Load Balancer (SLB), and Juniper strongly recommends the use of the internal SLB over an external SLB to improve system performance and reliability. This information is provided for users that are retrofitting the DX appliance behind an existing SLB.

If you use an external Server Load Balancer (SLB), you can use the SLB to direct traffic to the DX appliance rather than the web servers without interrupting the flow of traffic to your site. The simple deployment means that the DX appliance can be gracefully introduced into and removed from service without interrupting traffic flow. This can be done both manually for maintenance needs, and automatically for hands-off failure recovery.

Follow these steps to integrate the DX appliance into your network without any site downtime:

1. Ensure that the DX appliance is serving pages from the target server. This can be done by accessing the DX appliance through a web browser and verifying that it is passing back pages from the target server.

2. Add the DX appliance to the list of servers to which the server load balancer is directing traffic. In this configuration, the web traffic flowing through the DX appliance will be accelerated and the web traffic flowing directly to the web server will not be accelerated.

3. Verify that the DX appliance is servicing some of the web requests by looking at the DX appliance server statistics. This can be done either through DXSHELL with the `show server stats 1` command or by looking at the DX appliance Stats page in the WebUI.

4. Once you are comfortable that the DX appliance is serving pages, re-direct all traffic bound for the web server(s) to the DX appliance.

Integrating the DX Appliance into a Direct Server Return (DSR) Environment

Overview

The DX appliance can be easily deployed behind a Server Load Balancer (SLB) in DSR environments with a minimum amount of configuration. This simple deployment means that the DX appliance can be gracefully introduced into and removed from service without interrupting traffic flow. This can be done both manually for maintenance needs, and automatically for hands-off failure recovery.

What is Direct Server Return (DSR)?

DSR allows web servers to bypass the load balancer when responding to requests. With DSR, the web server sends HTTP responses directly to the requesting client, hence the name “Direct Server Return”.

Why use DSR?

Because HTTP responses (i.e., page data, images, etc.) are much greater in size than requests, using DSR greatly reduces traffic flow through the load balancer.

How Does DSR Work?

In a conventional, non-DSR environment, the SLB replaces the destination IP address in each client request packet with the IP address of the optimal target web server.

With DSR, the load balancer does not modify the destination IP address of client request packets. Instead, the load balancer changes each request packet's destination MAC address to that of the target server. Each target web server is configured with a loopback IP address that matches the SLB Virtual IP address (VIP). This allows the target host to accept request packets from the SLB and generate response packets that can be sent directly to the client without modification.

Note that only the SLB responds to Address Resolution Protocol (ARP) requests for the VIP to ensure that the router only forwards client requests to the SLB. Also, because only the MAC address is changed, the load balancer and its target servers must reside on the same layer 2 network.

Inserting the DX Appliance into a DSR Environment

1. Configure a cluster on the DX appliance whose listen VIP matches the SLB VIP by typing the commands:

```
dx% add cluster
dx% set cluster <name> listen vip <VIP of SLB>
dx% set cluster <name> listen port 80
```

2. Add target web servers that are currently a part of your DSR configuration to the cluster by typing the commands:

```
dx% set cluster <name> target host <ip:port for target host 1>
dx% set cluster <name> target host <ip:port for target host 2>
dx% set cluster <name> target host all enabled
...
```

3. On the DX appliance, enable DSR for this cluster with the command:

    dx% **set cluster <name> dsr enabled**

4. Save and apply the changes with the command:

    dx% **write**

5. Configure the SLB to forward client requests to the DX appliance(s) instead of the pool of web servers by specifying the actual interface IP address instead of the VIP address.

6. Configure the SLB to use the actual web servers as a backup pool for the DX appliance unit(s).

If the DX appliance is taken out of service, the SLB will transparently direct traffic to the target web servers, returning the site to its prior non-accelerated performance level until the SLB brings the DX appliance back into the traffic flow.

CAUTION: Target servers should keep their loopback address configuration in order to allow them to handle DSR traffic should the DX appliance be taken out of service.

Client IP Transparency

DX Application Acceleration Platforms operate in a secure reverse-proxy mode. In this mode, all incoming client requests are terminated at the DX appliance and multiplexed to a pool of pre-defined target hosts that serve the content. When the DX appliance provides connection multiplexing, the Source IP (SIP) is replaced by the IP of the DX appliance before the request is forwarded to the target host. This is required to provide the connection multiplexing capability within the DX appliance. However, this may create unintended side-effects:

- The target host logs do not have the client's IP address any longer.
- Since all requests to the target host seem to originate from a single IP, the host may perceive the traffic as an attack and close the connection.

If you have an application that looks at the client's IP address, there are two ways around this problem:

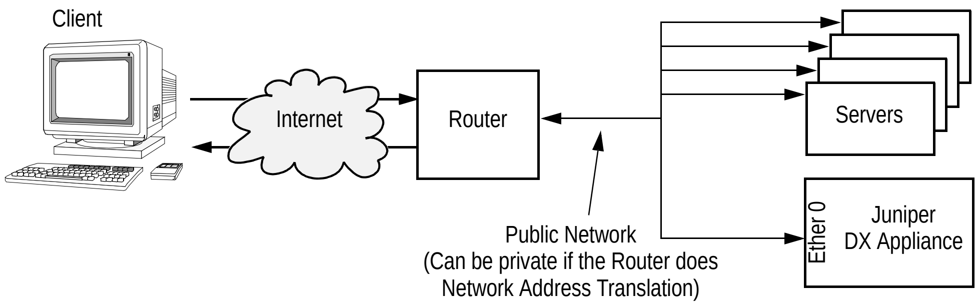
- Change your application to get the client's IP address from the Juniper "clientipaddr" header instead of the source address.

or

- Use the Client IP Transparency feature.
  - The Client IP Transparency feature allows you to enable or disable the client IP transparency capability for a cluster configuration.
- You will need to consider the following when Client IP Transparency is enabled:
  - The DX appliance will no longer off-load the server (the DX appliance does not pre-establish a session and multiplex client requests in the same persistent session).
- The target servers must use the DX appliance as their default route.

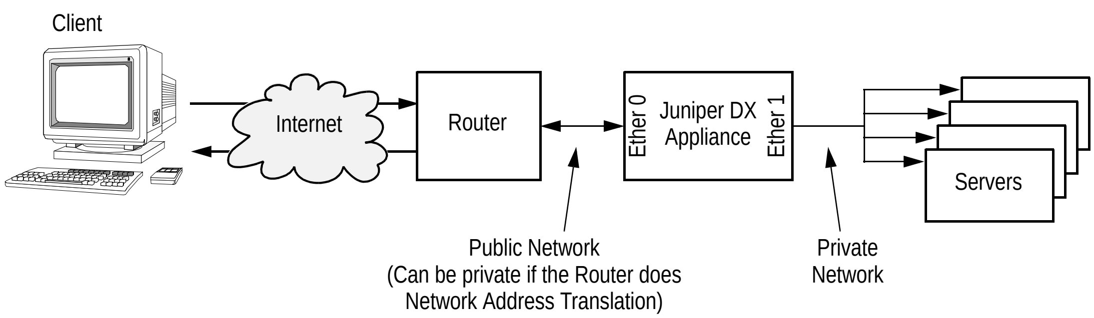
The target hosts must be located on a local subnet directly accessible by the DX appliance, and the clients must come from remote subnets. In the "One-Arm" topology (Figure 27), the DX appliance Ether 0 port and the web servers must be on one subnet, and the clients must be on other subnets. If there are only a handful of clients, this requirement can be circumvented by using static routes on the server for each client.

Figure 27: One Arm Topology



In “In-Line” mode (Figure 28), the DX appliance Ether 1 port and web servers must be on one subnet, and the DX appliance Ether 0 port and clients must be on other subnets.

Figure 28: In-Line Topology



NOTE: Client IP Transparency does not support traffic originating from the target hosts and passing through the DX appliance to any remote destination. Contact your Juniper Service Representative if you have any questions or concerns.

Client IP Transparency Commands

Transparency is disabled by default. This allows the DX appliance to operate in the normal manner. Enabling or disabling IP transparency will take effect only after a `write` operation.

The DXSHELL command used to set Client IP Transparency is:

```
dx% set cluster <name> transparency [enabled | disabled*]
```

The user must be Administrator or Network Administrator to use the `set` command.

The DXSHELL command used to show Client IP Transparency status is:

```
dx% show cluster <name> transparency
```

An Administrator, Network Administrator, Network Operator, or a user may use the `show` command.

Source Network Address Translation

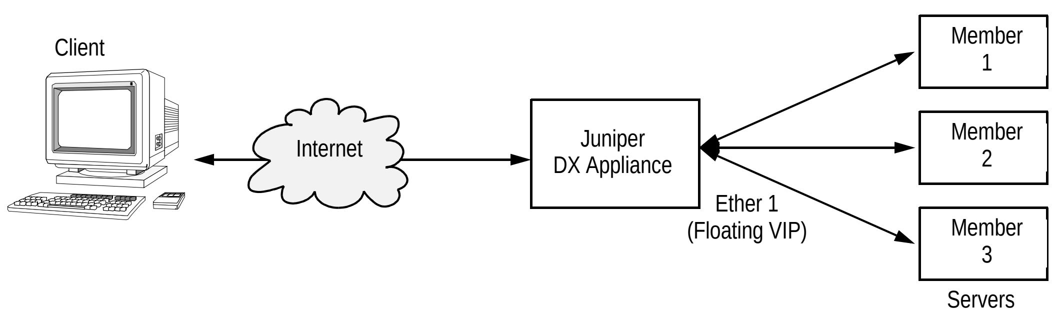
Source Network Address Translation (SNAT) provides external internet access to servers sitting behind a DX appliance. This becomes critical when the server has the DX Application Acceleration Platform IP configured as the default gateway, as in the case of Client IP Transparency. SNAT translates the server's source IP address to the Virtual IP address of either the SLB or a cluster. The SNAT feature is akin to a simple DSL router. The reverse traffic is converted back to its original IP address and sent back to the server. Currently only many-to-one conversion is allowed.

SNAT Operation

Currently three IP protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) are supported. When a DX appliance receives a packet from a member matching the IP and net mask, it associates it with a session and the source IP and port are replaced with the VIP and a chosen port. The reverse is done on the return traffic from the extranet and connectivity to external network is achieved.

Figure 29 shows the basic operation of SNAT. Servers (called members here) have a floating VIP configured as their default gateway. When a packet matching the member IP and net mask is received on the DX appliance, network translation is applied to the packet by replacing the source IP from the member IP to a VIP and sent via Ether 0, assuming that the default gateway of the DX appliance exists on Ether 0. If that is not the case, then the interface address of the interface facing the default gateway is chosen. Also if the chosen VIP is not aliased on the Ether 0 Interface, then the interface IP address is chosen.

Figure 29: Basic Operation of SNAT



SNAT Configuration Commands

Add Commands

To add a new SNAT group, type the command:

```
dx% add snat group [name]
```

The name is optional. If a name is not provided, a name starting from 1 will be allocated.

Delete Commands

To delete a SNAT group, type the command:

```
dx% delete snat group <name>
```

Set Commands

To add a new member to a group, type the command:

```
dx% set snat group <name> newmember [name]
```

The name is optional. If a name is not provided, a name starting from 1 will be allocated.

To set the VIP for the group, type the command:

```
dx% set snat group <name> vip <IP>
```

To set a group member's IP address, type the command:

```
dx% set snat group <name> member ip <IP>
```

To set the member mask, type the command:

```
dx% set snat group <name> member mask <MASK>
```

To set the maximum number of connections, type the command:

```
dx% set snat maxconn
```

The default is 1000 connections. The minimum is 1 and maximum is 1000.

To set the maximum idle time, type the command:

```
dx% set snat idletime
```

The default is 500 seconds. The minimum is 1 second and maximum is 24 hours.

Clear Commands

To remove a member from the group, type the command:

```
dx% clear snat group <name> < member | all >
```

Show Commands

To display all information related to the Source Network Address Translation (SNAT) configuration., type the command:

```
dx% show snat
```

To display the maximum number of connections, type the command:

```
dx% show snat maxconn
```

To display maximum idle time, type the command:

```
dx% show snat idletime
```

To display group information, type the command:

```
dx% show snat group [name | all]
```

To display members in group, type the command:

```
dx% show snat group <name> member < name | all >
```

To display the VIP for a group, type the command:

```
dx% show snat group <name> vip
```

To display the IP address for a member of a group, type the command:

```
dx% show snat group <name> member <name> ip
```

To display the netmask for a member of a group, type the command:

```
dx% show snat group <name> member <name> mask
```

Floating VIP

When using Client IP Transparency, the server sees the IP address of the actual client. In order to allow the responses go through the DX appliance, the server must have one of the DX appliance IP addresses configured as its default gateway. This configuration has a problem in that if the server is on a different network than that of the client, the DX appliance is forced to use the IP address of the interface facing the server (Ether 1, for example).

When a failover occurs in this condition, the server still has its default gateway pointed to the old active unit (which is currently down or passive). This causes the traffic to go to the wrong unit. The solution to this problem is to use a "Floating VIP." A floating VIP is a VIP that floats between two units in failover and always remains on the active unit.

A floating VIP is used on the interface facing the servers so that the floating VIP is always aliased in the active unit. This way the server always finds the correct unit as the default gateway. The floating VIP must be based in the same subnet as the server.

To add a floating VIP, type the command:

```
dx% add floatingvip <ip>
```

To delete a floating VIP, type the command:

```
dx% delete floatingvip <ip | all>
```

To show all of the floating VIPs, type the command:

```
dx% show floatingvip
```

Connection Binding and Microsoft's NTLM Authentication Protocol

The DX appliance improves application server capacity by multiplexing requests over a few persistent connections to the server farm to conserve the target servers' resources. In some environments, it is necessary to bind a connection from the user to the target server instead of allowing user requests to use an arbitrary connection to the target server. Multiplexing of connections may potentially allow an authenticated connection to be used by non-authorized users, violating the security policy.

Environments that use the NT Lan Manager protocol (NTLM) for authentication to Microsoft Proxy Servers require connection binding. NTLM is a proprietary protocol that authenticates connections rather than users or requests. Therefore, multiplexing connections to the target server must be disabled to avoid violating the NTLM authentication scheme.

Configuring Connection Binding

The connection binding feature provides the option of binding a connection from a single client to a target server. Connection binding is off by default, and can be enabled on a cluster-by-cluster basis.

1. To enable client to target server connection binding:

   dx% **set cluster <name> connbind enabled**

   In addition, you should configure the following for optimum performance.

2. Enable client IP-based client “stickiness” (refer to “Setting up the DX Appliance for “Sticky” Traffic” on page 155 for additional information).

3. Ensure that the web server keeps connections alive by setting a long connection time. The suggested value is five minutes or more.

4. To disable the following factory-set server settings:

   a. Disable the addition of an HTTP warning header by typing:

   dx% **set server factory h w disabled**

   b. Disable adding or appending to the HTTP Via header by typing:

   dx% **set server factory h v disabled**

   c. Close the connection to the target server when a 304 response is received by typing:

   dx% **set server factory h tc3 disabled**

Connection Binding and Layer 7 Health Checking

When L7 health checking is enabled and the target servers are NTLM-enabled, the expected HTTP return code of the health check should be set to 401 instead of the default of 200. Because the health check connections from the DX appliance to the target servers are not NTLM authenticated connections, health check requests return 401 “Unauthorized” instead of 200 “OK”. The DX appliance can make sure that the web server is up and running, but access to content is denied due to the non-authenticated connection.

To set the expected return code to 401, enter the command:

```
dx% set cluster <name> health returncode 401
```

Reverse Route Return

With “Reverse Route Return”, the DX appliance automatically adds routes when packets come back from a node that does not already appear in the DX appliance’s routing table. The problem reverse route return solves is that it is possible to lose packets when there is more than one gateway and the default gateway is not where the packet originated. Reverse route return allows response packets to be sent to the router that originally sent the request packets. This is done automatically, without the user having to manually configure static routes (a very time-consuming, error prone procedure).

For example, assume that there are two routers in the network (R 1 and R 2) and R 1 is the default gateway. If the DX appliance receives a packet from R 2 and there are no routes configured for the particular destination, the response will be routed towards R 1.The information that the request (or original packet) came from R 2 instead of R 1 is not included. Reverse route return remembers the path where the request was originated and enables the DX appliance to send the packet back to the router from which it was received.

Behavior

Normally, when a packet arrives on aDX Application Acceleration Platform from a node, it is not guaranteed that the response to the packet will go back to the same node. This is because of the way routing works in the operating system. Routing does not “remember” the node from which it received the packet. Instead the routing module decides where the packet should go by using current entries its routing table. In cases where there is no explicit routing entry for the destination, the packet is sent to the default gateway.

If the original packet did not arrive from the default gateway, but instead arrived from a different route, the response may not reach the actual destination. One way to counter this problem is to have the user configure static routes to the destination manually, but this method is not only time-consuming, but also error prone. Also at times it may not be possible to predict the path that a packet will take before arriving at the DX appliance. The solution to this problem is whenever a packet is received in the system, it is checked for following cases:

- Did the packet originate from another network?
- The incoming packet is from the DX appliance’s default gateway.

- There is not a static route configured on the DX appliance for the source IP of the incoming packet.

If all the above conditions apply to the incoming packet, then a route is created to the destination with the next-hop being the node from which the DX appliance received the packet. It is similar to adding a route manually from the command line, except that here the route does not stay indefinitely.

The DX appliance times-out the routes based on the activity. If a route is not used in the number of seconds specified by the `set server reversepath timeout` command, it is removed. This removes stale routes, freeing up memory in the DX appliance. The user can also limit the number of routes that can be added by this method using the `set server reversepath maxroutes` command.

When adding routes, the next-hop is derived from the ARP table and matched against the source hardware address (MAC address) in the packet. There can also be a case where there is not an ARP entry for the MAC address. This is usually the case if the system has recently been rebooted and all the ARP information has been reset. In this case, the DX appliance will change the destination MAC address when it sends the response packets out.

Reverse Route Return Commands

The following commands are provided for configuring the reverse route return feature. This command enables or disables the feature. The default value is disabled.

```
dx% set server reversepath < enabled | disabled >
```

To configure the maximum number of routes that can be added, use the command:

```
dx% set server reversepath maxroutes < number >
```

The minimum number is one, the maximum is 500, and the default value is 20

To configure the maximum timeout value for the entries added, use the command:

```
dx% set server reversepath timeout < secs >
```

The routes will be deleted after this interval of inactivity. The minimum value for timeout is one second, the maximum is 5000 seconds, and the default is 45 seconds.

Any settings made by the `set server reversepath` command will only take effect following the `write` command.

To display the current configuration of the reverse route return feature, use the command:

```
dx% show server reversepath
```

To display the current maximum number of routes that are allowed, use the command:

```
dx% show server reversepath maxroutes
```

To display the current timeout value, use the command:

```
dx% show server reversepath timeout
```

To display the current entries created in the system, use the command:

```
dx% show server reversepath entries
```

To clear an entry created by reverse route return, use the command:

```
dx% clear server reversepath entry < ip >
```

TCP Selective Acknowledgement

Multiple packet losses from a window of data can have a catastrophic effect on TCP throughput. TCP uses a cumulative acknowledgment scheme where received segments that are not at the left edge of the receive window are not acknowledged. This forces the sender to either wait a round-trip time to find out about each lost packet, or to unnecessarily retransmit segments that have been correctly received. With this cumulative acknowledgment scheme, multiple dropped segments generally cause TCP to lose its ACK-based clock, reducing overall throughput.

Selective Acknowledgment (SACK) is a strategy that corrects this behavior in the face of multiple dropped segments. With selective acknowledgments, the data receiver informs the sender about all segments that have arrived successfully, so the sender need only retransmit the segments that have actually been lost.

The DX Application Acceleration Platform supports Selective Acknowledgment, and it is always on; no configuration is needed.

Configuring a Virtual LAN

A Virtual LAN (VLAN) is a network of computers that behave as if they are connected to the same physical network even though they may actually be located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible. One of the biggest advantages of VLANs is that when a computer is physically moved to another location, it can stay on the same VLAN without any hardware reconfiguration.

The DX appliance uses VLAN tagging to differentiate packets belonging to different VLANs in a multi-VLAN environment. VLAN tags are also useful when two switches with multiple VLANs configured are connected in tandem. This enables the switch to transfer packets safely between the switches and not spill it out of the VLAN.

The mechanics of VLAN tagging are described by IEEE 802.1q Standards. The VLAN tag is a two-byte value inserted between the hardware layer header (an Ethernet header in our case) and the network header (IP or ARP header). A VLAN tag is identified mainly by the 12-bit ID that is part of the VLAN header. It can have a value ranging from 1 to 4095.

The DX appliance inserts VLAN headers in the outgoing packets based on the destination address. It can also insert VLAN tags based on a range of destination addresses.

Behavior

To understand the behavior of the VLAN feature, you should understand these terms:

- TPID: The Tag Protocol Identifier is set to 0x8100 to identify the frame as a IEEE 802.1q tagged frame.
- PRIO: The Frame Priority field is used to prioritize the traffic.
- CFI: The Canonical Format Indicator is a one-bit flag that indicates that the MAC address is in canonical format.
- VID: The VLAN ID is a two-byte header added in between the Ethernet header and the network layer header. It contains the VLAN tag, which is a value between 1 - 4095. (It is a 12-bit number, where 0 and 4096 are reserved values.)

There are different ways in which the VLAN tagging feature can be implemented. The classical method involves creating virtual interfaces. Each VLAN tag is associated with a virtual interface that has it's own subnet and IP address. All the packets that originate from this interface will have the VLAN ID. Each virtual interface is linked with a physical parent interface. For example, this type of system could have a configuration like this:

```
vlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet 172.21.12.34 netmask 0xffff0000 broadcast 172.21.255.255
ether 00:02:b3:b2:cb:51
vlan: 10 parent interface: em0
```

In this case, vlan0 is a virtual interface that has an IP address of 172.21.12.34. The VLAN ID associated with this virtual interface is 10 and its parent interface is em0.

This method is very effective if the VLANs are separated at subnet boundaries. The basic guideline of this method say "if you are sending packet from this X interface, the packet will have Y VLAN tag". While this architecture is well-defined and tested, it is very inflexible. It does not allow you to have two target machines that are in same subnet to be in different VLANs.

To counter this issue, the DX appliance changes the philosophy from "all packets from this interface will have VLAN ID X" to "all the packets going from or to this IP address will have VLAN ID X." In other words, if the packet contains this IP as either the source or destination IP, then the tag is inserted. In case there is a conflict between the tag for the source IP and the tag for destination IP, the destination IP will take precedence. Subnet boundaries do not affect the VLAN tags.

VLAN Commands

These commands are used to configure the VLAN feature.

Set Commands

To add a VLAN tag, type the command:

```
dx% set vlan ip < ip > < tag >
```

or

```
dx% set vlan range < startip-endip > < tag >
```

A tag added with a specific IP address takes precedence over a range. For example, if you add:

```
dx% set vlan range 192.168.10.100-192.168.10.200 10
```

and

```
dx% set vlan ip 192.168.10.34  456
```

the tag will have a VLAN ID of 456 instead of 10, even though IP 192.168.10.34 falls in the specified range.

Clear Commands

To delete a VLAN entry when added as a single entry, type the command:

```
dx% clear vlan ip <ip | all>
```

For example:

```
dx% clear vlan ip 192.168.56.7
```

To delete a VLAN entry when added as a range, type the command:

```
dx% clear vlan range <startip-endip | all>
```

For example:

```
dx% clear vlan range 192.168.56.4-192.168.56.10
```

Note that this command works only on entries that were added as ranges.

To clear an entry based on the tag value:

```
dx% clear vlan tag <tag>
```

For example:

```
dx% clear vlan tag 10
```

Show Commands

To display VLAN tags in the system, type the command:

```
dx% show vlan
```

To show VLAN entries based on ranges in the system, type the command:

```
dx% show vlan range <start-end | all>
```

To display VLAN entries based on IP addresses, type the command:

```
dx% show vlan <ip|all>
```

Pausing a Target Host

Target Host Pausing allows you to move target host servers in and out of rotation as live web servers without forcing a restart of the DX appliance. Target host pausing allows you to place a particular target host into a “soft” or a “hard”-paused condition:

- Soft pausing halts all new client traffic to the target host, but allows all existing “in-use” traffic to continue indefinitely.

- Hard pausing halts all new client traffic to the target host, and terminates all existing in-use traffic.

This action is performed on the command line and takes effect immediately after you press the enter key. If you then issue a `write` command, the paused condition is saved in the configuration. Otherwise, it remains only a runtime change; restarting the server or rebooting the DX appliance will cause that behavioral change to be lost. A paused server can be taken out of the paused state by issuing an `unpause` command.

To avoid potential race conditions that result in undefined and/or aberrant behavior, the pause/unpause commands only take local effect on the in-memory configuration when the server is down. The user is then notified on the command line accordingly. For example:

```
dx% set server down
dx% set cluster 1 target host 192.168.14.223:80 hardpaused
Server could not be contacted. Change applied only to in-memory config.
```

Because the target host pausing works on a per-cluster level, but target hosts are singularly represented within the server, pausing a target host suffers from similar anomalies as Layer 7 health checking. If the same target host is shared across two clusters, then the paused (or unpaused) condition will apply to all clusters that include that target host. This is based upon which condition is present first in the configuration file for any situation where the configuration file is `read` (server up/down, server crash, reboot). An example should clarify this.

Assumptions

- Cluster 1 has target host 192.168.10.100:80

- Cluster 2 has target host 192.168.10.100:80

- The target host is enabled for both clusters.

- The server is running.

Scenario 1

- Action: Customer pauses the target host 192.168.10.100:80 for Cluster 2.
- Result: Target host 192.168.10.100:80 is paused for both Clusters 1 and 2.

Scenario 2

Action: same as Scenario 1, except the operator does a "`write`", "`server down`", or "`server up`".

- Result: Target host 192.168.10.100:80 is paused for both clusters 1 and 2 before `write`. After `server up,` the target host 192.168.10.100:80 is not paused since the paused condition is not configured for cluster 1 and it was the first cluster in the configuration.

If the first cluster in the list has a shared target host set as paused, then all other clusters will have it set as paused when the server is restarted. When a target host is marked as paused for any cluster, it is paused within the server for all clusters, without regard to ordering.

When a cluster has all of its target hosts in some combination of "down" or "paused," then the cluster is marked down and the Global Application Failover (GAF) functionality comes into play (either "blackhole," "finclient," or "redirect"). The target host pausing condition should not be considered a state of the target host, but a behavior; a target host can be both "up" and "paused", or both "down" and "paused."

Target Host Pause Commands

The commands for target host pausing are as follows.

To enable "hard" pausing, type the command:

dx% **set cluster N target host X hardpaused**

To enable "soft" pausing, type the command:

dx% **set cluster N target host X softpaused**

To disable pausing (either hard or soft), type the command:

dx% **set cluster N target host X unpaused**

Using a Local IP for Target Host Communication

A local IP address can be configured to be used for communication with the target hosts. The configuration is per cluster and forwarder. The same local IP address is used for communication with all the target hosts in a cluster or forwarder. The source IP for all the connections (e.g., client requests, health check requests) initiated by the DX appliance to the target hosts will be this local IP.

By default, the local IP address is not configured. When the local IP address is not configured, the local IP address used is the IP address of the interface through which target host communication takes place.

The following rules apply when using a local IP address:

- Local IPs cannot be one of the interface IPs
- Local IPs cannot be one of the cluster/forwarder VIPs
- Local IPs cannot be the administrator VIP
- Local IPs has to be on the same subnet
- Local IPs will also be used as the source IP for health check connections

When the target hosts in a cluster or forwarder and the local IP are not in the same subnet, the traffic from the DX appliance to the target hosts and target hosts to the DX appliance might follow different routes. It is preferable and the feature is most useful, when they are in the same subnet.

When a target host is in more than one cluster, the local IP used for communication with the target host is the local IP of the first cluster that has the target host and the local IP configured. If no cluster has the local IP configured, the interface IP is used.

When a target host is in more than one forwarder, the local IP used for communication with the target host is the local IP of the first forwarder that has the target host and local IP configured. If no forwarder has the local IP configured, the interface IP is used.

Local IP Configuration Commands

To set the local IP to be used for communication with all the target hosts in a cluster, type the command:

```
dx% set cluster <name> target localip <ip>
```

To set the local ip to be used for communication with all the target hosts in a forwarder, type the command:

```
dx% set forwarder <name> target localip <ip>
```

To remove the local IP setting for the cluster, type the command:

```
dx% clear cluster <name> target localip
```

To remove the local IP setting for the forwarder, type the command:

```
dx% clear forwarder <name> target localip
```

To display the local IP setting for the cluster, type the command:

```
dx% show cluster <name> target localip
```

To display the local IP setting for the cluster/forwarder, type the command:

```
dx% show forwarder <name> target localip
```

Enabling Target Server Compression

The DX appliance automatically compresses responses sent to web browser clients. This improves downstream bandwidth utilization (traffic going to web browser clients). However the DX appliance, in its default mode, does not do the same thing when communicating with the target servers. In fact, it specifically asks for un-compressed content from the target servers. This wastes bandwidth and introduces delays, especially when the target web servers are in a remote site. “Target Server Compression” enables decompression capability in the DX appliance so that it can process compressed HTTP responses from the target web server. This greatly reduces the bandwidth requirements on the link between the DX appliance and the target servers.

Decompression is used when a request for a cluster VIP is forwarded to the target web servers. When the client makes a request to the VIP, the browser might send a “Content-Encoding” header indicating that it accepts a compressed content (in either gzip or deflated formata). Since in its default state the DX appliance cannot handle compressed content, it strips this header when forwarding the request to the target web servers. Enabling target server compression causes the DX appliance to include a content-encoding header indicating to the target servers that the DX appliance can understand compressed content. This needs to be enabled on per-cluster basis since the target server might not be able to support compression efficiently, or in a LAN environment, it might not be necessary to compress the content.

The DX appliance could also have Page Translator Content (PTC) Application Rules (refer to “Application Rules Syntax” on page 245 for additional information) enabled that modify the content. In this case, the DX appliance decompress the content so that PTC rules can be run. After the rules have been run, the response might need to be encoded again. The client compressed encoding format is “chosen” based upon the configuration and browser support.

Some browsers have a configuration called 2K padding, and do not accept compressed data unless a 2K block of compressed empty spaces precedes the compressed content. When 2K padding is set (using the command `set server compression 2k_padding enabled`), the DX appliance needs to decompress the content and pad it before compressing it again.

In some cases it might be preferable to forward the content-encoding header sent by the browser. This could help in cases where the browser understands an encoding that is supported by the target web server, but not by the DX appliance. Since the DX appliance does not understand this encoding, it needs to forward the header to the server rather than rewriting it based on what it supports. This is also

useful in cases where the target web server knows more about the client and wants to send the encoding it prefers.

This means that when the response comes back, the DX appliance must not modify the response. In this case, PTC rules will not be run, so a warning message is displayed in DXSHELL and WebUI when the “Application Rule” is configured. To be consistent, PTC rules also will not be run if the response is un-compressed. (It is possible to run PTC on un-compressed content but this would mean some responses would have PTC run while others would not.) This is referred to as “Target” mode where the target knows the best content encoding to deliver to the browser.

For completeness, two more options are provided for this scenario: Standard mode, and Target Enhanced mode. Standard mode is as described above. In Standard mode, PTC rules can be run.

Target Enhanced mode allows the DX appliance to handle un-compressed content using the standard logic for compression, but still let compressed content through. This allows the DX appliance to add value in cases where the target web server cannot handle compression of certain content-types. To be consistent, in the Target Enhanced mode, PTC rules are also run on the response.

Previous versions of the DX appliance provided a similar option called “Pass-Accept Encoding”. This was controlled using the command `set server compression pass_ae`. The same is now achieved by setting the Processing mode to target and encoding to browser.

Caching is also affected by this feature. The DX appliance caches responses based upon the encoding, and in theory, it could cache multiple encodings for the same content. When a request is received, the DX appliance looks up the response in the cache based upon the “chosen” compression encoding. This choice is controlled by configuration and is part of the logic of the Standard mode.

This presents a problem when the DX appliance is sending browser encoding and needs to follow Target or Target Enhanced modes. In this case, caching this response is not useful as the DX appliance cannot utilize it properly. If it were to cache the response, it could not determine on the subsequent request which encoding to fetch from the cache.

In these modes, the DX appliance needs to be explicitly told that the target knows more than the DX appliance, and it needs to allow the target to decide on the encoding. In this case the only thing the DX appliance can cache is un-compressed content. This applies to both Target and Target Enhanced modes. The DX appliance can safely cache un-compressed response since in most cases the content will be gzip/images that will not be compressed by the either the target web server or the DX appliance. And the DX appliance will not see cases where the target web server compresses the same content sometimes and not others. With Standard mode, the DX appliance can safely cache any response since in this case the DX appliance is being instructed to follow its own logic to choose the encoding for the client (and therefore, can look up the proper encoding in the cache).

Table 5 shows the allowable configuration combinations and caching/PTC characteristics.

Table 5: Configuration Combinations and Caching/PTC Characteristics

Encoding/Mode	None	Standard	Target	Target Enhanced
Standard	X	X		
Browser	X	X	X[1]	X[1]

1.No PTC rules run and only uncompressed responses are cached.

Setting the configuration for a processing mode of None, Standard, Target and Target Enhanced controls whether the DX appliance asks for compressed content from the target web-server and what logic to follow in dealing with the response. Selecting "None" disables the feature.

Setting the configuration for encoding to standard or browser controls which content-encoding header to send.

Target Server Compression Commands

This section shows the commands used to control target server compression.

Set Commands

To set the Target Server Compression mode, type the command:

dx% **set cluster <name> compression targetcompression mode [none | standard | target | target_en]**

The options are:

- None: No target compression (default). If this is set then the "encoding" configuration is not considered.
- Standard: Target compression is enabled. Perform standard processing; applicable to both standard and browser encoding configurations.
- Target: Target compression is enabled. Any responses from the target are not touched. Encoded responses are not cached, but un-encoded responses are cached; PTC is not run. This is only applicable for encoding configuration of the browser.
- target_en: Target compression is enabled. Encoded responses from the target are not touched. Un-encoded responses are compressed following the "standard" logic for compression. Encoded responses are not cached but un-encoded responses are; PTC is not run. This is only applicable for encoding configuration of the browser.

To set the Target Server Compression encoding method, type the command:

dx% **set cluster <name> compression targetcompression encoding [standard | browser]**

The options are:

- Standard: The DX appliance "Content-Encoding" header is sent. Standard encoding is applicable in either None or Standard modes (default).

- Browser: “Browser Content-Encoding” header is sent. “Browser-Encoding” is applicable in the modes None, Standard, Target, and Target Enhanced.

The `set` commands require administration rights of the Administrator or the Network Administrator.

Show Commands

To show Target Server Compression mode, type the command:

```
dx% show cluster <name> compression targetcompression mode
```

To show the Target Server Compression encoding method, type the command:

```
dx% show cluster <name> compression targetcompression encoding
```

To show decompression statistics, use the commands:

```
dx% show cluster <name | all> stats http
dx% show cluster <name> target host <host> stats http
dx% show server stats http
```

To show the historical stats for decompression, use the commands:

```
dx% show cluster <name> stats history http target decompression [performed | failure]
[hour | day | month | year]

dx% show cluster <name> target host <host> stats history http target decompression
[performed | failure] [seconds | minutes]

dx% show server stats history http target decompression [performed | failure] [hour |
day | month | year]
```

Target Server Compression under the WebUI

The cluster WebUI screen has a section titled “Advanced: Compression.” In it, there is the option entitled “Target Compression” mode with the choices of [None | Standard | Target | Target Enhanced], and an option entitled “Target Compression Encoding” with the choices of [Standard or Browser]. The DX appliance and Cluster Stats sections of the WebUI have a “Decompression” section that contains the items “Decompression Performed” and “Decompression Failure.”

Chapter 7

Forward Proxy Accelerator

This chapter describes the Forward Proxy Accelerator for the DX Application Acceleration Platform, discussing the following topics:

Overview

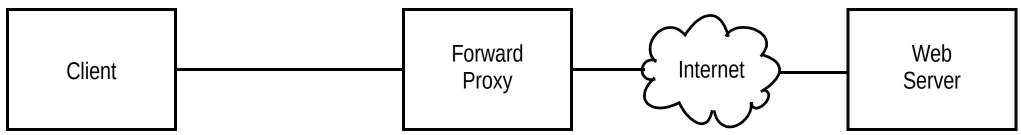
The Forward Proxy Accelerator enables the DX Application Acceleration Platform to accelerate HTTP traffic served by a forward proxy. The DX appliance itself is NOT the forward proxy. From a position in front of a forward proxy, the DX appliance transforms normal HTTP requests (i.e., GET, POST, PUT, etc.) as usual using compression, OverDrive (AppRules), etc. The DX appliance also detects HTTP CONNECT requests from clients, and forwards data on those connections between the client and the forward proxy without any transformation.

The Forward Proxy Accelerator is an optional feature that requires a license file to work. Contact your Juniper Networks Sales Representative to obtain a license.

Forward Proxy Background Information

This section provides background information on HTTP browser connections to forward proxies. Assume that the forward proxy only listens on port 80. (In the Internet Explorer this is set up under Tools- > Internet Options- > Connections- > LAN Settings- > Proxy Server). Here are three common scenarios for forward proxy exchanging HTTP traffic, based upon the network setup as seen in Figure 30.

Figure 30: Forward Proxy Network Setup

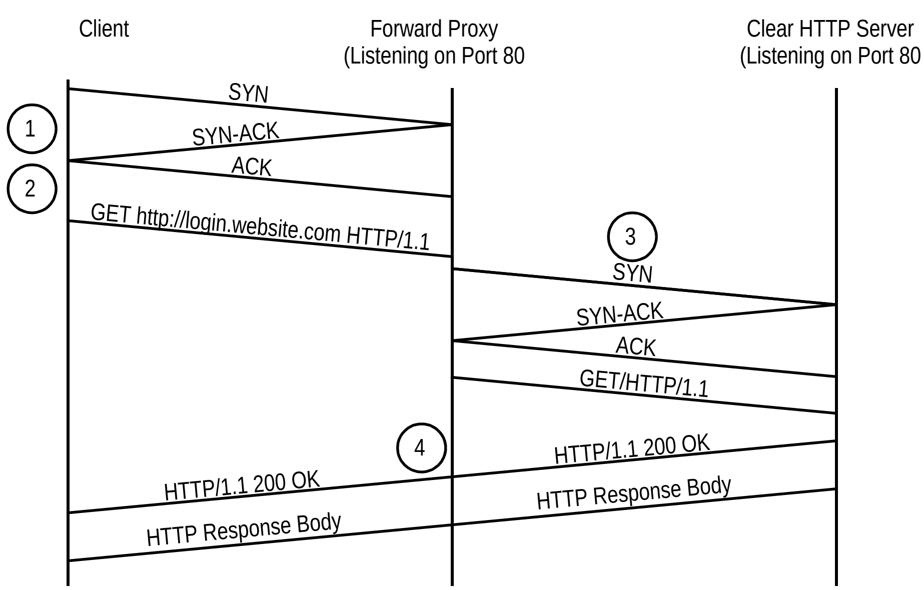


Clear Request for a Clear Page

This scenario is used by browsers to retrieve clear pages through a forward proxy (refer to Figure 31).

1. The client establishes a TCP connection to port 80 of the forward proxy.
2. The client sends a “GET http://login.website.com HTTP/1.1” request for a clear (non-SSL) page to the forward proxy. Note that the URL includes the “http://”.
3. The forward proxy uses DNS to resolve “login.website.com” to an IP address, establishes a TCP connection to port 80 of that IP address, and sends “GET / HTTP/1.1”.
4. The response from the web server is forwarded back to the client. The forward proxy can manipulate the HTTP headers as needed.

Figure 31: Clear Pages through a Forward Proxy



CONNECT Request for a Secure Page

This scenario is used by browsers to retrieve SSL pages through a forward proxy (refer to Figure 32).

1. The client establishes a TCP connection to port 80 of the forward proxy.

2. The client sends "CONNECT login.website.com:443 HTTP/1.1" to the forward proxy. For example:

```
CONNECT login.website.com:443 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.5)
Gecko/20041107 Firefox/1.0
Proxy-Connection: keep-alive
Host: login.website.com
```

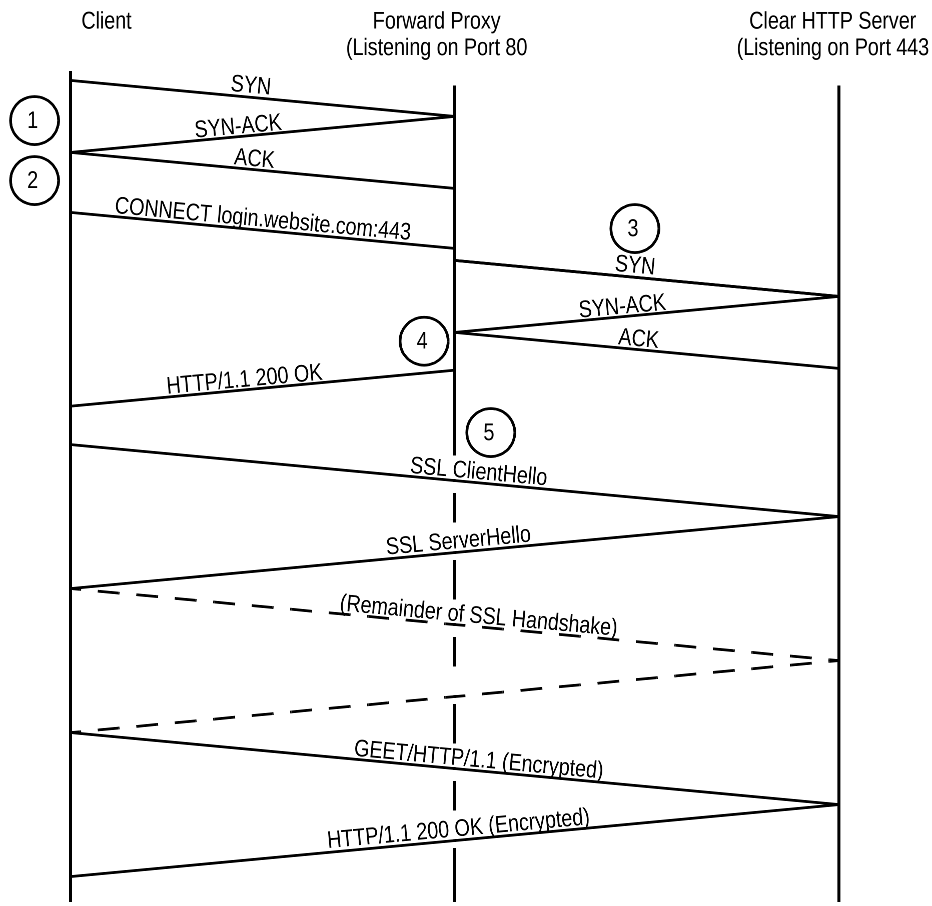
3. The forward proxy uses DNS to resolve "login.website.com" to an IP address, and establishes a TCP connection to port 443 of that IP address.

4. The forward proxy sends back a "Connection Established" response to the client. For example:

```
HTTP/1.0 200 Connection established
Proxy-agent: Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6e
```

5. At this point, the client establishes an SSL connection on the existing TCP connection (i.e., by exchanging ClientHello and ServerHello messages, etc.), but the other endpoint of the SSL connection is the web server at "login.website.com", not the forward proxy. The forward proxy simply forwards bytes back and forth between the client and login.website.com, but does not, and cannot, decrypt the application data.

Figure 32: SSL Pages through a Forward Proxy



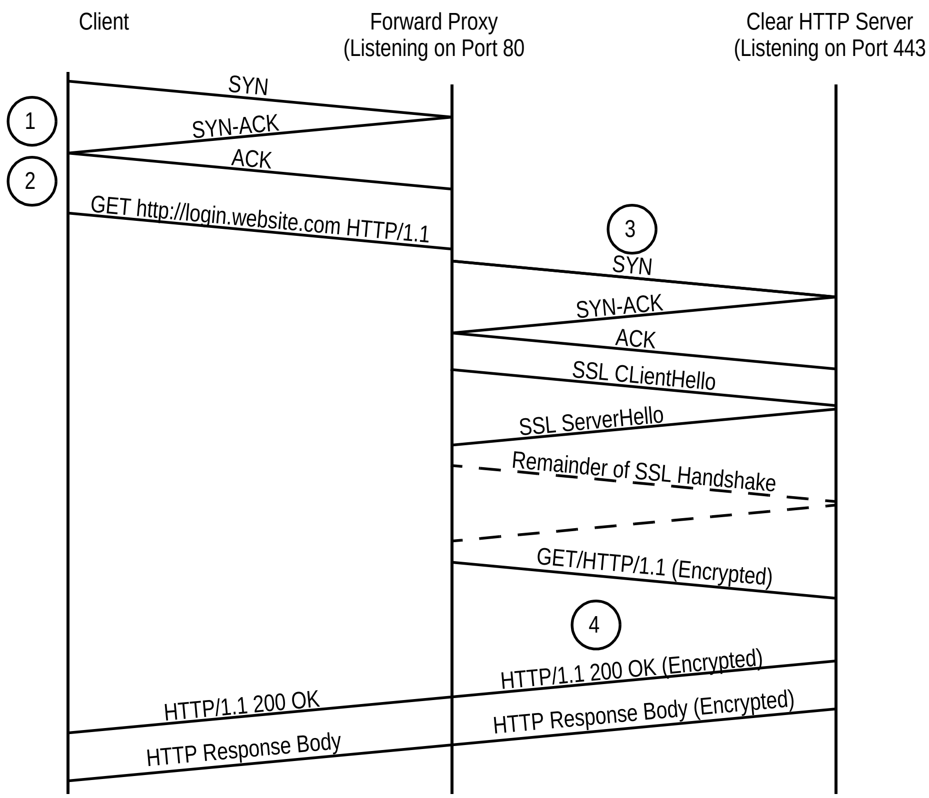
Essentially, the CONNECT method allows tunneling of other TCP-based protocols (like SSL) over HTTP. The CONNECT method is available in all HTTP versions (0.9, 1.0, 1.1).

NOTE: In this scenario, the forward proxy cannot inspect or modify the application data.

Clear Request for Secure Page (without CONNECT)

This scenario does not seem to be used by browsers, probably for security reasons, but it may be used by custom clients. It is mentioned here because it would be the only way that the forward proxy could inspect and modify the application data for SSL connections.

Figure 33: Clear Request for a Secure Page (without CONNECT)



1. The client establishes a TCP connection to port 80 of the forward proxy.

2. The client sends a "GET https://login.website.com HTTP/1.1" request for an SSL page to the forward proxy.

3. The forward proxy uses DNS to resolve "login.website.com" to an IP address, establishes a TCP and an SSL connection to port 443 of "login.website.com", and sends "GET / HTTP/1.1".

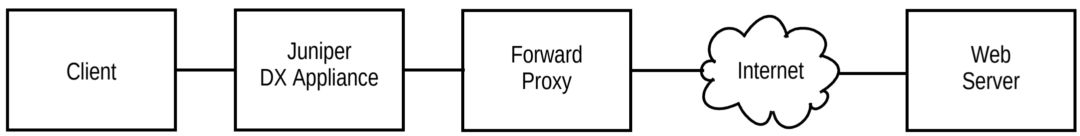
4. The SSL response from the web server is decrypted and forwarded back to the client in the clear. The forward proxy may manipulate the HTTP headers.

A variation on this scenario is to also have an SSL connection between the client and the forward proxy, but a different SSL connection than the one between the forward proxy and the web server.

Forward Proxy with the DX Application Acceleration Platform

With the Forward Proxy Accelerator feature, the DX appliance can sit between the client and the forward proxy. Previously, if the DX appliance was located in front of a forward proxy, it could only support scenarios (1) and (3). This feature allows the DX appliance to sit in front of the forward proxy and support the connect method in scenario (2) as well. Figure 34 shows a diagram of the network setup.

Figure 34: Forward Proxy Network Setup

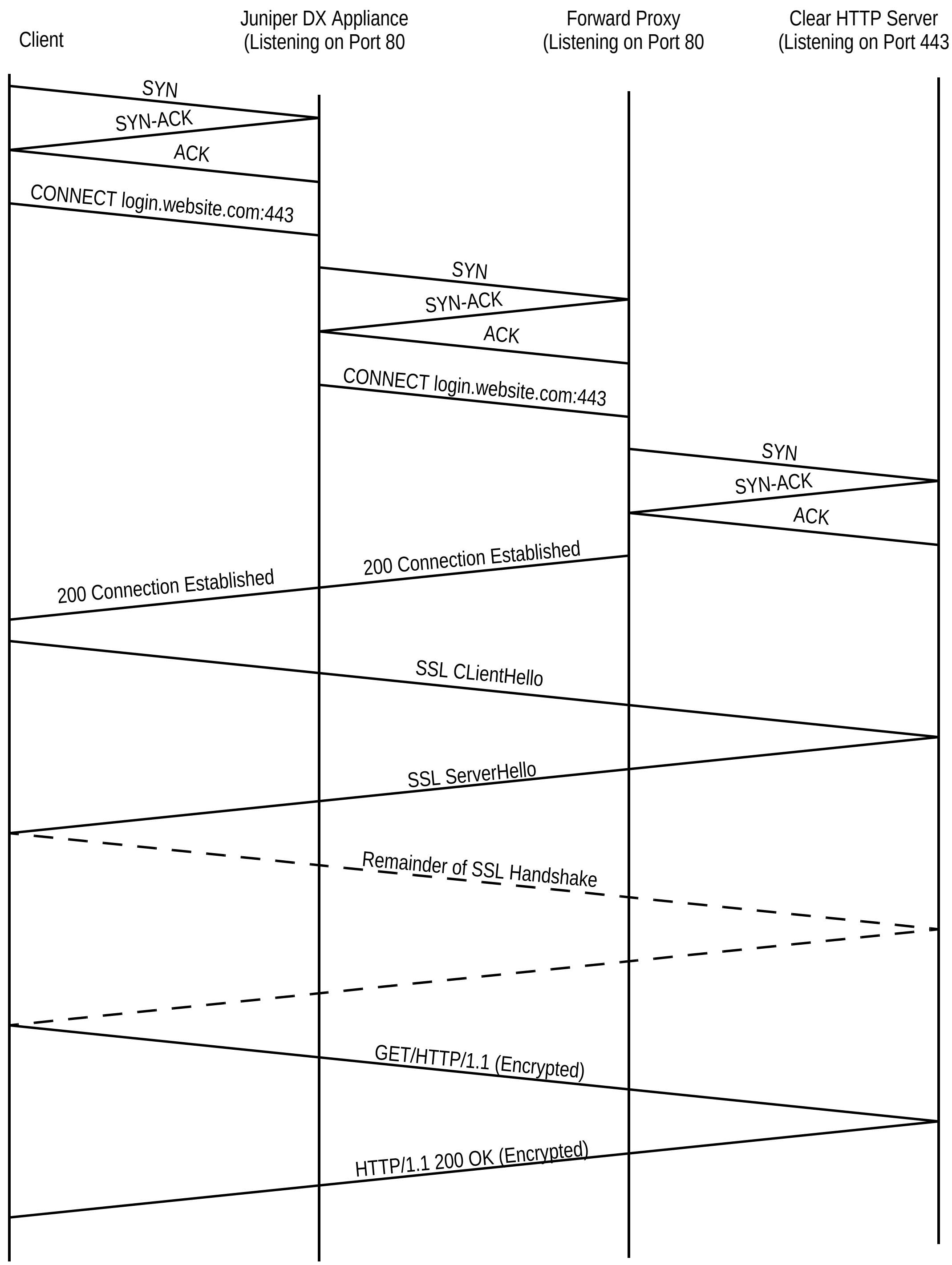


Connection binding is required to ensure that target sessions for CONNECT method requests are not reused.

Figure 35 shows a diagram of the connection setup for the CONNECT method.

NOTE: Because browsers primarily use scenarios (1) and (2), the DX appliance will only be able to accelerate clear traffic and not SSL traffic.

Figure 35: Forward Proxy with DX Application Acceleration Platform CONNECT Method



Forward Proxy Accelerator User Interface

This section describes the user interface for the Forward Proxy Accelerator.

Command Line Interface Commands

Target Tuning

Use the `Target Tuning` command to enable the Forward Proxy Accelerator feature. An example:

```
dx% set cluster 1 target tune

This will help optimize the communication with the Target Hosts
    within this cluster. It will help ensure that functionality is
    maintained while providing the most possible benefit.

    Please answer the following questions. Enter Control-C at any
    time to exit without modification ('*' denotes default selection).

    1) Please select the Target Application
        1) Other (*)
        2) PeopleSoft
        3) Domino5
        4) Domino6
        5) JDE
        6) OWA
        7) Fwd Proxy

    Enter Selection: 7

    2) Please select the Target Web Server Type
        1) Other (*)
        2) Apache
        3) IIS4

    Enter Selection: 1

    You have selected:
        Target Application: Fwd Proxy
        Target Web Server: Other
        NTLM Authentication: Required

    Continue using these selections?
        N) No, Start Over (*)
        Y) Yes, Use these values

    Enter Selection: Y

    Tuning based on your selections ...

    Done.
    (*) dx% write
```

The audit log will display an entry as:

```
[2005-03-18 17:07:49 (+0800)] local [juniper] [cli] cluster “1” target tune:
Application = Fwd Proxy, Server = Other, NTLM = Required
```

Cluster Set Commands

Alternatively, the Forward Proxy Accelerator feature can be enabled with these commands:

Connection Binding: Connection binding is required because target sessions that handle SSL traffic via the CONNECT method cannot be reused. Enable connection binding by typing the command:

```
dx% set cluster <name> connbind enabled
```

HTTP CONNECT: Enable support for the CONNECT method by typing the command:

```
dx% set cluster <name> httpmethod connect enabled
```

Forward Proxy Accelerator with the WebUI

Target tuning has not been added as yet to the WebUI for any target type. On the WebUI, the user can manually enable “Connection Binding” and “HTTPMETHOD CONNECT”.

Chapter 8
Configuring for High Availability

This chapter describes Configuring for High Availability for the DX Application Acceleration Platform, discussing the following topics:

Overview

The DX Application Acceleration Platform performs health checks on target hosts. When target hosts fail, the DX appliance can route the traffic to other available target hosts. However, if the DX appliance is deployed in a standalone mode, and the DX appliance stops responding because of network errors (or for any other reason), the target hosts (and therefore, the web site or applications) may be unavailable to the client until the issues are resolved.

The DX appliance can be deployed in three different topologies to increase system availability that will be discussed in detail in other sections:

- Active-Standby
- Active-Active
- ActiveN

To provide you with a better understanding of the various topologies, the "Glossary" on page 365 will provide a series of terms that will be used in the explanations.

This chapter provides on overview of the three different topologies used to increase system availability. If you would like detailed information on how high availability configurations work, refer to "Layer 4 Switching and ActiveN" on page 375.

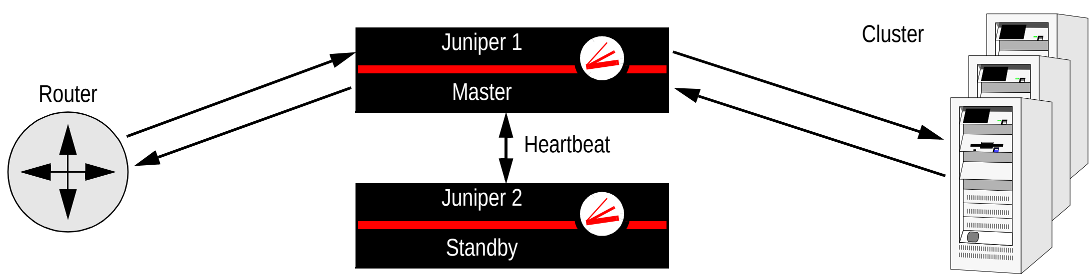
Topologies

Active-Standby Topology (Active One)

Active-Standby Configuration is a two-appliance configuration where one DX appliance processes client traffic and load-balances the client requests (the active unit) while the other (standby unit) listens to the active unit's heartbeat and waits to take over as the active unit in case the active unit fails. The heartbeat is sent between the two the DX appliances using Ether 0 (default) or another bind address if configured.

In the event of failure of the active unit, the standby unit detects the failure within five seconds, and then takes over as the active unit. This heartbeat interval is configurable. During the takeover, the standby DX appliance broadcasts gratuitous ARP messages to advertise that it now owns the Virtual IP and the Virtual MAC address previously associated with the active unit. This causes any upstream routers to recognize the new interface ports and route subsequent client requests to the standby (now the active unit).

Figure 36: Active-Standby Topology

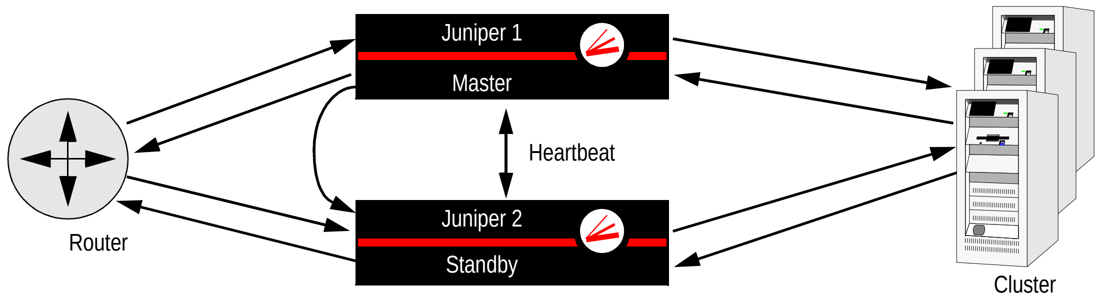


While the active-standby topology is an effective way of assuring high availability of the site, it is not an efficient use of the DX appliances because only one of them is processing requests at any one time. An active-active or ActiveN topology is the recommended approach. For additional information on how to configure an active-standby system, refer to "Achieving High Availability and Failover with Active-Standby Topology" on page 117.

Active-Active Topology

The Active-Active Configuration is a two-appliance configuration where both DX appliances are actively processing client traffic and load balancing the client requests. One of the DX appliances is the token "Master" and if the Master fails, the remaining DX appliance takes up the Master role, taking and redistributing requests from clients. The ActiveN topology is recommended for high availability and high scalability over active-standby configurations.

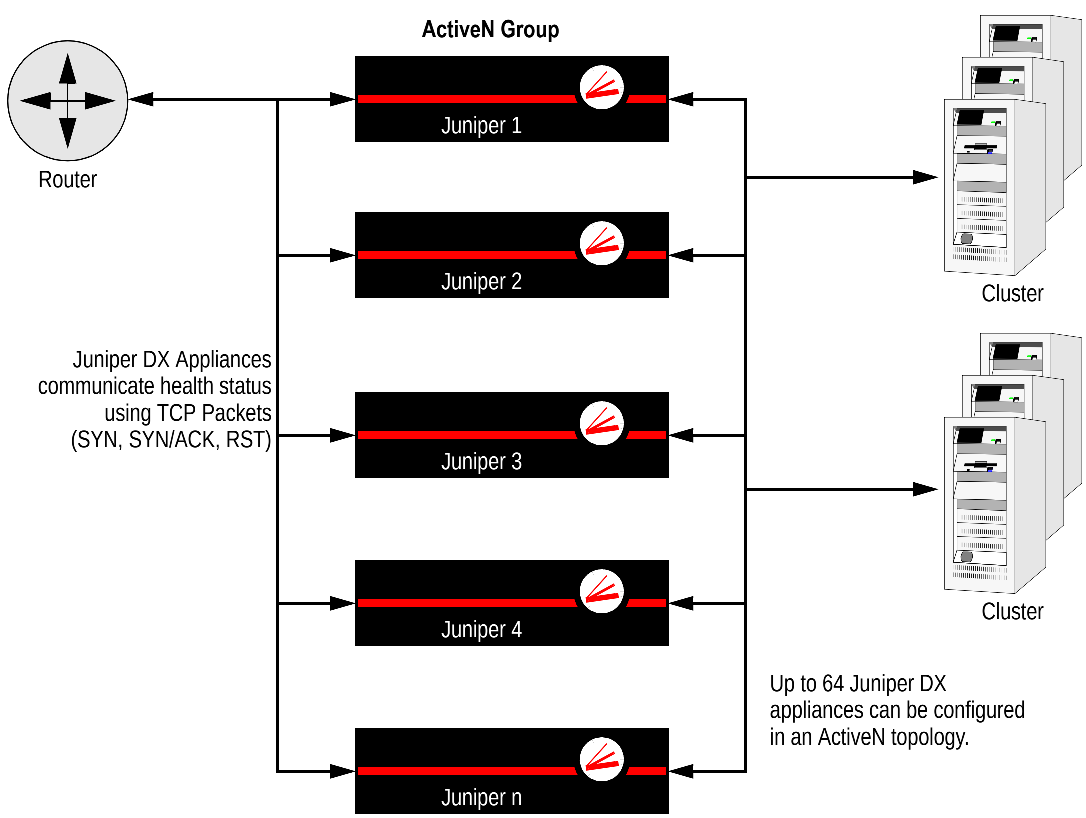
Figure 37: Active-Active Topology



ActiveN Topology

ActiveN is an extension of the active-active topology that allows scaling of the network. ActiveN allows up to 64 (N) DX Application Acceleration Platforms to actively process traffic destined for a VIP without the need for an external "Server Load Balancer" (SLB); refer to Figure 38. This allows horizontal scaling of DX appliances to process multiple gigabits of outbound response data, and enable configurations that are highly resistant to failure.

Figure 38: ActiveN Topology



ActiveN ensures that all operational DX appliances continue to process traffic regardless of how many (or which) peer units are lost or disabled. ActiveN is used in network configurations where multiple DX Application Acceleration Platforms are deployed. Any one of the DX appliances can be the Master (or active unit) that takes the requests from clients and redistributes the traffic to the rest of the DX appliance.

If the Master DX appliance fails, one of the remaining DX appliances takes up the Master role, taking and redistributing requests from clients. ActiveN is based upon the Layer 4 switch functionality built into the DX appliance.

Achieving High Availability and Failover with Active-Standby Topology

To configure a failover unit, you must enable a “Failover” mode on two DX appliances configured with the same VIP address. The DX appliance will automatically determine which unit is active and which is the failover based upon the order in which failover is enabled on the two units. The DX appliance that has failover enabled first will become the primary unit. The DX appliance with failover enabled second will monitor the first DX appliance. If the active DX appliance goes offline, the failover DX appliance will activate and take its place.

1. Ether 1 has a default IP address configured and you can optionally configure with a different IP. Make sure that the Ether 1 Heartbeat ports on both the primary DX appliance and the standby DX appliance are connected to your switch. If instead you are connecting the pair via crossover, be sure to set the media for both ports to 10/100/1000BaseT.

2. OPTIONAL: Because failover uses multicast, the factory default IP and netmask should work without any changes, but you can provide valid IP and Netmask settings for both DX appliances Ether 1 Heartbeat interfaces. This can be done from either the DXSHELL command line or through a web browser with the WebUI.

   a. From the DXSHELL

      You can view interface settings from the DXSHELL with the command:

```
dx% show ether 1
```

      You can change interface settings from the DXSHELL with the commands:

```
dx% set ether 1 ip <address>
dx% set ether 1 netmask <netmask>
```

      where `<address>` and `<netmask>` are the IP Address or Netmask values you wish to enter.

      To save changes, use the command:

```
dx% write
```

To view and change interface settings for Ether 1 in the WebUI, open the Network Settings page. Be sure to click the SAVE button to save any changes.

1. Make sure that the primary DX appliance and the failover DX appliance have the same Virtual IP address.

   a. From the DXSHELL

      You can view VIP settings from the DXSHELL with the command:

```
dx% show cluster <name> listen vip
```

      You can change VIP settings from the DXSHELL with the command:

```
dx% set cluster <name> listen vip <IP Address>
```

NOTE: If the VIP and IP addresses for the DX appliance are on the same subnet, then the VIP netmask must be set to 255.255.255.255.

      To save any changes, use the command:

```
dx% write
```

   b. From the WebUI

      To view and change VIP settings in the WebUI, open the DX appliance Settings page and edit the settings of the cluster whose VIP you would like to change. Be sure to click the SAVE button to save any changes.

2. OPTIONAL: Enable failover with Virtual MAC (VMAC)

   With this option enabled, the active unit will use the Virtual MAC as its MAC address. The VMAC ID determines the virtual MAC address. The default for the VMAC ID is zero. When a standby unit becomes active, it will assume the VMAC as its MAC address.

   a. Enable failover with the VMAC

```
dx% set server failover vmac enabled
```

   b. Configure the VMAC ID

```
dx% set server failover vmac id <VMAC ID>
```

3. The active DX appliance will shut down in the event of an Ethernet link failure.

   When the Ethernet link is determined to be down, the amount of time before the active DX appliance will shut itself down is the POLL INTERVAL * COUNT. This value should be less than three seconds.

   Poll interval is a value in milliseconds. The default poll interval is 500 milliseconds. The count is an integer, and the default count is 4.

4. Enable failover on BOTH the primary DX appliance and the failover DX appliance.

    a. From the DXSHELL

        You can enable failover from the DXSHELL with the command:

    dx% **set server failover enabled**

    b. From the WebUI

        To enable failover, open the Administrator Services page and locate the "High Availability Failover" option and check "Enabled". Be sure to press the SAVE button to save and apply the changes.

5. To activate failover from the DXSHELL, use the following command on both servers:

    dx% **set server up**

    Determination of which server is the active server and which is the backup is negotiated by the DX appliances, based upon the network address (the unit with the higher network address becomes the active unit).

Initiating a Manual Failover

There are times when you will need to take a server off-line for maintenance or debugging purposes. You can initiate a manual failover in an active-standby configuration on the active server by typing the command:

dx% **set server down**

When the backup server fails to detect the heartbeat messages coming from the active server, it takes over processing and becomes the active node.

Either the server that you took off-line or a replacement unit can be returned to activity as a backup unit by typing the command:

dx% **set server up**

For example, if you modify a configuration on a cluster on an active DX appliance that requires a restart of the multiplexing engine, the process brings the active DX appliance down, and the standby DX appliance takes over and becomes the active DX appliance. This may result in a web site not processing requests while the standby DX appliance takes over. If you want to make configuration changes to an active-standby configuration without affecting request processing, use the following sequence:

1. Ensure that DX appliance 1 and DX appliance 2 are in an active-standby configuration (DX appliance 1 is active and DX appliance 2 is the standby).

2. Change the cluster configuration on DX appliance 2 (passive).

3. Move the traffic to DX appliance 2 (set the server down on DX appliance 1).

4. Check the failover status on DX appliance 2 (now active).

5. Check the ActiveN status on DX appliance 2 (now active).

6. Bring DX appliance 1 up (set the server up on DX appliance 1).

7. Check the failover status on DX appliance 1 (now passive).

8. Check the ActiveN status on DX appliance 2 (now standby).

9. Change the cluster configuration on DX appliance 1.

Active-Active and ActiveN Configuration

The Active-Active Configuration is a two-appliance configuration where both DX appliances are actively processing client traffic and load balancing the client requests, and when one DX appliance dies, the other assumes the dead DX appliance's duties. ActiveN is simply an active-active topology with more than two DX appliances connected together. Think of it as "Active-Active-Active..." for both load balancing and failover. It is actually better than traditional active-active topology, not only because of its ability to support more than two machines, but because it also provides truly linear scalability in terms of performance.

Taking Advantage of ActiveN

Each machine you want to use with ActiveN needs to know that it is part of this "Active-Active-Active..." configuration. Since ActiveN performs both load balancing and failover, both will configure.

The steps for setting up all machines in the ActiveN group (refer to Figure 38) are identical because ActiveN automatically manages which machine is the "Master." Many of the details are automatically negotiated by the DX appliances.

You can use these instructions to quickly setup ActiveN on each of your machines.

Configuration Steps

Before you get started:

- These instructions assume your DX appliances are on the same subnet and are meant to be used with one another in a single group.
- These instructions also assume you will use Ether 0 on each DX appliance.
- Remember to replace items in angle brackets < > with your own settings.

Configuring DX Appliance 1

1. Create and populate your cluster. A cluster is a set of redundant web servers used to serve the same content. The cluster provides load balancing and redundancy for your content; ActiveN provides load balancing and redundancy for the machines that provide your clustering capabilities.

```
dx% add cluster <name>
dx% set cluster <name> listen vip <listen_vip>
dx% set cluster <name> listen port <listen_port>
```

```
dx% set cluster <name> target host <ip:port for web server 1>
...
dx% set cluster <name> target host all enabled
```

2. Set DSR. ActiveN uses (and actually requires) “Direct Server Return”. The performance benefits are substantial and it is important that you enable DSR on all ActiveN machines. You do not need to enable DSR on your web servers. This setting is transparent to your target hosts. For additional information on DSR, refer to “Integrating the DX Appliance into a Direct Server Return (DSR) Environment” on page 81.

   To enable DSR, type the command:

   ```
   dx% set cluster <name> dsr enabled
   ```

3. Create an ActiveN group. This group contains all of the DX appliances that you will use together in the ActiveN configuration. Link your cluster and ActiveN group by using the IP and port settings you used for the cluster in Step 1.

   ```
   dx% add activeN group <name> <listen_vip:listen_port>
   ```

4. Define the machines you will use. ActiveN calls each machine in the ActiveN group a “Blade”. First, you will create these blades, and then you will assign them to your ActiveN group.

   Create blades by referring to the real IP address of Ether 0 on the machine you want to add to the ActiveN group. You can repeat this step for each machine you want to add:

   ```
   dx% add activeN blade <rl1_real_ip>
   dx% add activeN blade <rl2_real_ip>
   ...
   dx% write
   ```

5. Now, assign the blades you created to your ActiveN group. Instead of doing this individually, use this provided shortcut:

   ```
   dx% set activeN group all blade all
   ```

6. Start ActiveN and your server. Remember that ActiveN has both load balancing and failover components.

   ```
   dx% set activeN failover enabled
   dx% set activeN enabled
   dx% write
   ```

This completes the ActiveN setup on the first computer. Here are a few items to remember:

- All clusters in the ActiveN group you created must all have DSR set to enabled, per Step 2. You need not change any settings on your target hosts.

- ActiveN will automatically determine the “Master” machine unless you specify otherwise using the “Forcemaster”. All DX appliances must have the same settings for the forcemaster (`set all to enabled` or `set all to disabled`) for ActiveN to work, so be careful if you are changing it.

- ActiveN will automatically assign node identification numbers, so setting the node ID manually is not covered in this guide. Remember though, that all DX appliances must all have different node ID settings.
- You can check your ActiveN configuration using the command:

  dx% **show activeN group <name | all>**

Configuring DX Appliance 2

Repeat Step 1 to Step 6 to configure DX appliance 2. Note that the cluster and activeN configuration parameters must be identical on the DX appliances, except for the node ID.

Adding More DX Appliances

If you want to add another DX appliance to ActiveN failover, simply repeat Step 1 to Step 6 on each of the new DX appliances. Similarly, you can also configure more ActiveN groups on DX appliances. For example, you can add a group as 10.0.22.12:443 and configure an SSL cluster on each DX appliance to serve HTTPS requests.

Making Changes After Configuring ActiveN

Remember that ActiveN has both load balancing and failover components, so if you would like to disable ActiveN while you are making configuration changes, it requires the following two commands:

```
dx% set activeN failover disabled
dx% set activeN disabled
```

Afterwards, while it is not required, you may want to stop your server while you are making changes:

```
dx% set server down
```

A helpful tip for preserving uptime: you can make your changes on Slave machines first. Force a failover, and only then make changes to the remaining machine (the one that used to be the Master). Use the following command to see which machine is acting as the Master and make edits to that machine last.

```
dx% show activeN failover
```

Finally, it is important to note that if you make changes to one DX appliance's ActiveN configuration, you must ensure that you make that change to ALL the DX appliances you are using for that ActiveN group. After you are finished, re-enable ActiveN load balancing and failover using the commands:

```
dx% set activeN failover enabled
dx% set activeN enabled
```

If you stopped your server, remember to bring to back up again:

```
dx% set server up
```

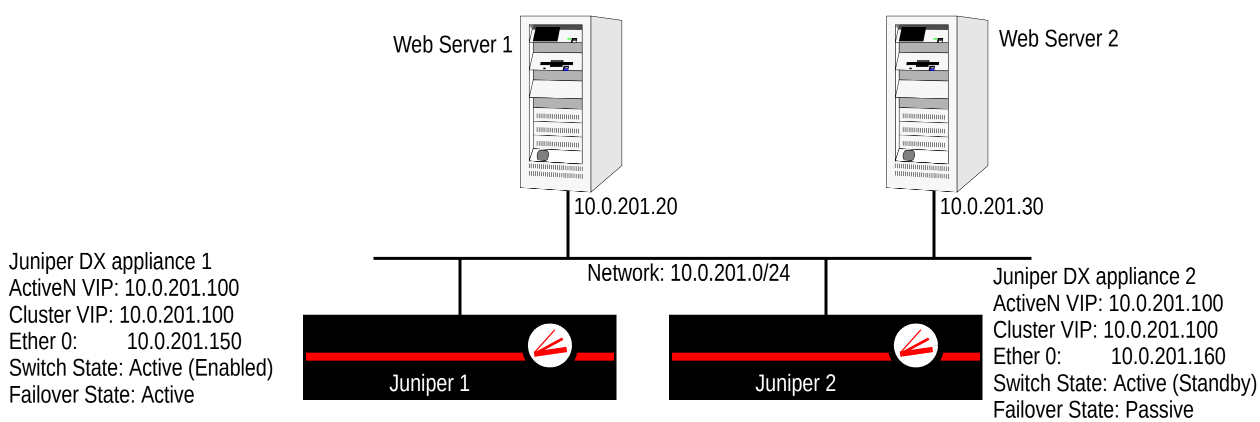
Sample ActiveN Configuration

Figure 39 shows an example of an ActiveN configuration, along with the commands needed to set up the configuration. For simplicities sake, this example only has two DX appliances. This network has the IP address map shown in Table 6.

Table 6: Example Network IP Address Mapping

DX Appliance	Port	IP Address
Web Server 1		10.0.201.20:80
Web Server 2		10.0.201.30:80
Juniper DX Appliance 1	Ether 0	10.0.201.150
	Ether 1	10.0.201.151
Juniper DX Appliance 2	Ether 0	10.0.201.160
	Ether 0	10.0.201.161
ActiveN VIP		10.0.201.100
Cluster VIP		10.0.201.100

Figure 39: An Example of an ActiveN Configuration



DX Appliance 1:

Cluster Configuration

```
dx% add cluster 1
dx% set cluster listen vip 10.0.201.100
dx% set cluster 1 target host 10.0.201.20:80
dx% set cluster 1 target host 10.0.201.20:80 enabled
dx% set cluster 1 target host 10.0.201.30:80
dx% set cluster 1 target host 10.0.201.30:80 enabled
dx% set cluster 1 dsr enabled
dx% write
```

ActiveN Configuration

```
dx% add activen group 1 10.0.201.100:80
dx% add activen blade 10.0.201.150
dx% add activen blade 10.0.201.160
dx% set activen group 1 blade 10.0.201.150
dx% set activen group 1 blade 10.0.201.160
dx% set activen failover enabled
dx% set activen enabled
dx% write
```

DX Appliance 2:

Cluster Configuration

```
dx% add cluster 1
dx% set cluster listen vip 10.0.201.100
dx% set cluster 1 target host 10.0.201.20:80
dx% set cluster 1 target host 10.0.201.20:80 enabled
dx% set cluster 1 target host 10.0.201.30:80
dx% set cluster 1 target host 10.0.201.30:80 enabled
dx% set cluster 1 dsr enabled
dx% write
```

ActiveN Configuration

```
dx% add activen group 1 10.0.201.100:80
dx% add activen blade 10.0.201.150
dx% add activen blade 10.0.201.160
dx% set activen group 1 blade 10.0.201.150
dx% set activen group 1 blade 10.0.201.160
dx% set activen failover enabled
dx% set activen enabled
dx% write
```

ActiveN Commands

These are the commands used to configure ActiveN.

Set Commands

`Set` commands are used to set configuration parameters.

Global Configuration Commands

This command is used to turn-on or turn-off the ActiveN feature by setting the switch state.

```
dx% set activeN <enabled|disabled>
```

The switch can be set in one of two states:

- Enabled: The switch is active.
- Disabled: The switch is stopped.

This command is used to set the cleaning interval (the interval at which two repeat cycles are spaced):

```
dx% set activeN cleaning_interval <secs>
```

Since the ActiveN switch works in DSR mode, it does not see the packets going from blade to the client. This makes it difficult for the ActiveN switch to track the connection state. Instead, it uses a timer to purge the sessions.

```
dx% set activeN session timeout   active <secs>
dx% set activeN session timeout   closewait <secs>
dx% set activeN session timeout   ackwait <secs>
```

If a session has not been active for a period of time, it is purged in the timer. The three possible conditions are:

- Active: The session that is in active session.
- Closewait: The session the client has terminated from its side.
- Ackwait: The 3-way TCP handshake not completed.

This command is used to set the maximum allowable blades in the system.

```
dx% set activeN max_blades <number>
```

This command needs to be run before starting the ActiveN switch.

```
dx% set activeN advanced policy <roundrobin|leastconn>
```

This is used to set the switching policy to either round robin or least connection.

This command is used to enable protection against syn flood. Since the ActiveN operates in DSR mode, it cannot track if the 3-way TCP handshake completed successfully, but it needs to remember the session information for such sessions. In

order to protect itself from a attack, the ActiveN purges a connection if the client does not send final acknowledge for the handshake.

```
dx% set activeN advanced synflood_protect <enabled|disabled>
```

Since all the sessions are purged in the timer routine, we can set the maximum number of timed out sessions to be purged in one timer interval.

Setting burst_max to zero will cause all the sessions that have timed out to be purged in timer cycle.

```
dx% set activeN advanced burst_max <number>
```

This command is used to enable or disable the sending of resets to the client. When active sessions are purged, a reset can be sent to the client and to the server to indicate the connection has been terminated.

```
dx% set activeN advanced reset client <enabled|disabled>
```

This command is used to enable or disable sending of resets to the server (“blade”).

```
dx% set activeN advanced reset server <enabled|disabled>
```

This command is used to set the switch state.

```
dx% set activeN <enabled|disabled>
```

The switch can be set to one of two states:

- Enabled: The switch is started.
- Disabled: The switch is stopped.

Set Group Commands

This command is used to set a blade as a member of a group. Using the keyword “all” in the group argument results in the blade being added to all the groups, and using “all” in the blade argument results in adding all the blades into the group.

```
dx% set activeN group <name|all> blade <ip_addr|all>
```

This command is used to set the Client IP Sticky, which is where the load balancer chooses the same server for multiple TCP connections when the subsequent requests come from the same client IP address. Refer to “Client IP Sticky” on page 382 for additional information.

```
dx% set activeN group <name|all> blade sticky <enabled|disabled>
```

This command is used to set the timeout value for the Client IP Sticky feature. The default value is 120 minutes, the minimum is one minute, and the maximum is 30 days.

```
dx% set activeN group <name|all> blade sticky timeout <minutes>
```

Health Check Commands

Periodic health checks of the blades are conducted for status of the blades. The following commands set the parameters associated with the health checking. Note that Health check is a default feature and it cannot be turned-off.

These commands are used to set the time duration between two health checks.

```
dx% set activeN healthcheck interval up <secs>
dx% set activeN healthcheck interval down <secs>
dx% set activeN healthcheck interval syn <secs>
```

Intervals between two health checks are defined for each different status of the blades.

- `up`: The blade has responded to the health check probe.
- `down`: The blade has not responded to the probe and has been taken out of rotation.
- `syn`: Time gap between sending two consecutive health probes, if no response is received.

This command is used to set the maximum number of health check tries before giving up.

```
dx% set activeN healthCheck maxtries <Number>
```

The default values for each of these parameters are:

- Up: 45 seconds
- Down: 20 seconds
- Syn Wait: 10 seconds
- Maxtries: 3

NOTE: The default values for health checking are not optimum for all DX appliances. Refer to “ActiveN Health Checking Parameters” on page 135 for additional information.

Set Failover Commands

Failover commands are used to instruct the DX appliance what it should do when an error occurs.

This command is used to enable or disable the "Forcemaster". Enabling the forcemaster allows a switch to snatch "activeness" from another switch with a higher node ID.

```
dx% set activeN failover forcemaster <enabled|disabled>
```

This command is used to set the multicast address for the failover mechanism.

```
dx% set activeN failover mcastaddr <Ip addr>
```

This command is used to set the bind address for the failover mechanism.

```
dx% set activeN failover bindaddr <Ip addr>
```

This command is used to set the node ID of the ActiveN failover unit. Setting the node ID to auto will result in the node ID being generated automatically.

```
dx% set activeN failover nodeid <number|auto>
```

This command is used to set the port for failover communication.

```
dx% set activeN failover port peer <port>
```

This command is used to disable or enable the Virtual MAC (default is disabled).

```
dx% set activeN failover port vmac [disabled | enabled]
```

This command is used to assign the Virtual MAC Address to the specified ID.

```
dx% set activeN failover port vmac <id>
```

Set Client IP Sticky Commands

To enable Client IP Stickyness, type the command:

```
dx% set activeN group <name> sticky enabled
```

To disable stickyness, type the command:

```
dx% set activeN group <name> sticky disabled}
```

The default value of sticky is disabled.

To set the timeout of sticky entries, type the command:

```
dx% set activeN sticky timeout < minutes>
```

The default timeout is 120 minutes, the minimum is 1 minute, and the maximum is 30 days.

These commands require a role of Administrator or Network Administrator before they can be executed. A `write` is required in order for the changes to take effect.

Add Commands

This command is used to add a new group with optional name and VIP and port.

```
dx% add activeN group [name] <vip:port>
```

This command is used to add a new blade with a real IP; an index is returned.

```
dx% add activeN blade <Real IP>
```

Delete Commands

This command is used to delete a group specified by name. Using `all` will delete all groups.

```
dx% delete activeN group <name|all>
```

This command is used to delete a blade specified by an index. Using `all` will delete all blades.

```
dx% delete activeN blade <ip_addr|all>
```

Clear Commands

This command is used to disassociate a blade from a group. Using all will remove all the blades from the groups.

```
dx% clear activeN group <name|all> blade <index|all>
```

This command is used to clear the statistics for a group.

```
dx% clear activeN group <name|all> stats
```

This command is used to clear the statistics for a blade.

```
dx% clear activeN blade <name|all> stats
```

This command is used to clear overall statistics.

```
dx% clear activeN total stats
```

Show Commands

This command is used to display the group characteristics. Using all will display all of the groups.

```
dx% show activeN group <name|all>
```

This command is used to display the blade characteristics. Using all will display all of the blades.

```
dx% show activeN blade <ip|all>
```

This command is used to display the overall statistics for the switch.

```
dx% show activeN stats
```

The ActiveN statistics are cumulative for all running ActiveN groups. The statistics displayed are shown in Table 7.

Table 7: ActiveN Statistics

Statistic	Description
Total Statistics	
Bytes	The total byte count received by all clients.
Packets	The total number of packets received by all clients.
Flushed	The total number of connections that have been flushed by ActiveN. Once the DX appliance receives a RST or a FIN from the client for an active connection, it then waits a number of seconds, and flushes the connection. The counter is then incremented.
syn	The total number of SYNs sent by all clients.
rst	The total number of RSTs sent by all clients.
fin	The total number of FINs sent by all clients.
Current Sessions	
Active	The current number of established TCP sessions.
Fin	The current number of FINs sent by the client prior to ActiveN flushing.
Reset	The current number of RSTs sent by the client prior to ActiveN flushing.

Troubleshooting ActiveN problems depends upon the nature of the problem that is occurring. For instance, if the “active” session count is high and increasing, but the “flushed” count is low and not increasing, this implies that there are either slow clients/target hosts, or a high latency on transactions with the DX appliances.

By knowing what these values mean, you can keep track of what is going on in your site (primarily from the client side to the DX appliance). Dividing these numbers by time can give you the average occurrence count for each variable in the ActiveN stats.

This command is used to display the basic configuration parameters.

```
dx% show activeN
```

This command is used to display advanced configuration parameters.

```
dx% show activeN advanced
```

This command is used to display the switch state.

```
dx% show activeN status
```

This command is used to display the Client IP Sticky timeout entries.

```
dx% show activeN sticky timeout
```

For complete information on each of these commands. refer to the *Command Line Reference* manual.

Instant Redirect

Instant Redirect is a simple mechanism used to divert traffic from a cluster where all target hosts are down (i.e., a "dead" cluster) to an active cluster somewhere else in the network (world). The instant redirect feature allows a user to configure the cluster to respond with a redirect (HTTP 302 reply) instead of operating in the customary blackhole mode. When the cluster is completely down (due to all target hosts being down), new connections arriving at the cluster will have a 302 response sent to them immediately (the DX appliance does not even wait for the request to arrive). The response is made in HTTP 1.0 fashion with the connection being closed after sending out the response. This allows the DX appliance to respond at very high speed and rapidly reflect traffic to the new destination.

The instant redirect feature is configured using the command:

```
dx% set cluster <name> listen targetsdown [blackhole|finclient|redirect <url>]
```

where:

- `blackhole` refers to the current behavior of dropping all packets sent to the cluster that has all of its target hosts down.
- `finclient` refers to the historical behavior of allowing the client to connect and then subsequently closing down the connection with a FIN.
- `redirect <url>` refers to the new behavior of redirecting clients with an HTTP 302 reply to the new location specified in the `<url>`. The URL is specified as follows:

  http://<server>[:port][/path/resource]

To view the current configuration, use the command:

```
dx% show cluster <name> listen targetsdown
```

NOTE: The Instant Redirect feature only works with HTTP clusters, not HTTPS.

Connectivity Failover

Another method of achieving reliability is through the use of “Connectivity Failover”. Connectivity failover (also known as “Gateway Failover”) allows you to initiate failover in the active-standby topology by doing health checks on pre-configured hosts (ActiveN is supported). You must first set up a DX appliance in an active-standby topology. You then configure a group of IP addresses to health check; these IP addresses will be checked for layer 3 connectivity. If a health check fails, the standby DX appliance will take over as the primary.

WARNING: Enabling this feature and having a failover event causes the DX appliance to reboot.

WARNING: Be certain that the hosts you choose to health check are pingable when you add them into your health checking. If they are not, then the following occurs:

- Assume that DX appliance A is the active unit and DX appliance B is the standby unit. If you add the IP of a host that is down to remote host health checking, DX appliance A will not be able to ping that host, and will eventually failover to DX appliance B. This causes a reboot of DX appliance A.
- Depending upon the configuration of DX appliance B, it is possible that DX appliance A will not be done rebooting by the time DX appliance B reboots. This can cause significant problems.

The “Server Load Balancer” (SLB) can also be included in the failover (you must enable SLB failover). The configured IP addresses are checked for Layer 3 connectivity via a ping; no Layer 4 check is performed. If a configured number of health checks fail, the active DX appliance is switched to Standby mode. This allows the other DX appliance to become active.

To enable connectivity failover, you must enable failover with `set server failover enabled`. The following items must be configured:

- The health check interval (default value is 10 seconds; range: 10-600 seconds).
- The timeout value waiting for a response (default value: 10 seconds; range: 1-60 seconds).
- The maximum health check attempts per host (default value is 5; range: 1-60).
- The minimum remote hosts failing before activating failover (default value is 1; range: 1-10).
- The hosts to health check (maximum: 10).
- Connectivity failover enabled/disabled (default value is disabled).

The DX appliance will add an entry to the system logs when failover occurs. If the “timeout value” is larger than the “health check interval,” and a remote host is not responding, a health check request will not be sent until the “timeout value'” has expired. In other words, if a host is not responding, the health check interval becomes the maximum value of the interval and timeout.

If “minimum remote hosts failing” is larger than number of remote hosts, failover will never occur. The DXSHELL prints a warning when this condition is set.

The failover algorithm is:

> When the number of consecutive health check failures equals the “maximum health check attempts,” the host is considered down.
>
> After this, if the number of hosts down is equal or greater than the “minimum remote hosts failing,” failover will occur.

When failover is invoked, the active DX appliance will be rebooted to allow the standby unit to takeover. This feature will not startup unless one of the following is true:

- Active-standby failover is enabled
- ActiveN failover is enabled
- SLB failover is enabled

One of these must be enabled AND the DX appliance must be the active unit for connectivity failover to work.

Connectivity Failover Commands

Use these commands to configure connectivity failover.

To add an IP address to health check, type the command:

dx% **set health remotehost host [ip]**

To enable connectivity failover, type the command:

dx% **set health remotehost [enabled | disabled]**

To set the health check interval (how often to send the health checks), type the command:

dx% **set health remotehost interval [seconds]**

To set the health check timeout (how long to wait for a response), type the command:

dx% **set health remotehost timeout [seconds]**

To set the health check maximum number of attempts before considering the host down, type the command:

dx% **set health remotehost retry [count]**

To set the count for the minimum number of hosts failing, type the command:

dx% **set health remotehost minhostsfailing [count]**

To remove an IP address from health check, type the command

```
dx% clear health remotehost host [ip]
```

Show Commands

Use these show commands to see the status of connectivity failover:

```
dx% show health remotehost host
dx% show health remotehost status
dx% show health remotehost interval
dx% show health remotehost timeout
dx% show health remotehost retry
dx% show health remotehost minhostsfailing
```

These commands may be executed by the Administrator, Network Administrator, and Network Operator.

ActiveN Health Checking Parameters

Using the ActiveN default health checking parameters can cause the ActiveN "active" unit to forward traffic to non-healthy blades. A non-healthy blade is a cluster with "all" target hosts down. As long as there is at least one target host up in a cluster; then ActiveN will consider that cluster/blade a healthy blade.

Worse Case Scenario for ActiveN Forwarding Traffic to a Non-Healthy Blade

With the default health checking parameters (Up: 45, Down: 20, Syn wait: 10, and Maxtries: 3), ActiveN will distribute traffic to non-healthy blades for a period of up to 1 minute,15 seconds (the "worse case" scenario).

Best Case Scenario for ActiveN Forwarding Traffic to a Non-Healthy Blade

In the "best case" scenario in terms of time, traffic will still be forwarded to a non-healthy blade for a period of at least 20 seconds.

In some applications this is too long of a period of time to be forwarding traffic into a blackhole, and the default ActiveN health checking parameters are not aggressive enough. The health check parameters are configurable for applications that need quicker health checking results.

Suggested Values

A more aggressive ActiveN health checking parameters configuration might be:

- Up: 4 seconds
- Down: 4 seconds
- SYN wait: 2 seconds
- Maxtries: 1 seconds

Keep in mind that these configurable numbers can even be lower. However, in a healthy environment three packets (SYN, SYN ACK, and RST) are continuously exchanged between the ActiveN active unit and the blades that it is health checking. This packet exchange can become very chatty in ActiveN scenarios where there are multiple ActiveN blades and multiple ActiveN groups.

Chapter 9

Layer 7 Health Check

This chapter describes Layer 7 Health Check for the DX Application Acceleration Platform, discussing the following topics:

Layer 7 Server Health Checking with the DX Appliance

The DX appliance can perform Layer 7 (L7), content-based health checking for your target web servers. L7 health checking allows the DX appliance to examine content from a target host to determine if it is correctly handling requests. The DX appliance will stop sending client requests to a server that is having problems, resuming only once the target host has passed a specified number of successful health checks. L7 health checking is disabled (by default) and configured on a per-cluster basis.

When a target host is assigned to more than one cluster, the health check settings for the first cluster are used, and health check settings for the second and subsequent clusters are ignored. You should use the same L7 health check configuration for all clusters that contain the same target hosts. Various user-configurable aspects of the expected response are described below.

Health checking can also be extended to an SMTP server. In this method, the DX appliance establishes a TCP connection with the SMTP server and sends an initial handshake message (HELO). If the server responds with a valid response (a response code of 250), then the server is marked “up”. If not, the server is marked “down”. The same timeouts used for health checking of other ports also apply to SMTP health checking.

The SMTP health checking method of does NOT work with Secure SMTP. It works only with plain-text SMTP servers.

Health Check Settings

You can view a summary of L7 health check settings for a cluster by typing the command:

```
dx% show cluster <name> health
```

This is an example of output from the command along with explanations of what each parameter means and how to make changes from the command line:

```
dx% show cluster 1 health
Health Check Status: enabled
Health Check Interval: 150
Health Check Retry: 4
Health Check Resume: 1
Health Check Url Path:
Health Check Return Code: 200
Health Check Size: -1
Health Check String:
Health Check Timeout: 15
Health Check User Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0;
T31
dx%
```

Status

Status relates to whether or not L7 health checking is enabled.

Turn-on health checking with the command:

```
dx% set cluster <name> health enabled
```

Turn-off health checking with the command:

```
dx% set cluster <name> health disabled
```

Turn-on SMTP health checking with the command:

```
dx% set slb group <name | all> healthcheck smtp enabled
```

Turn-off SMTP health checking with the command:

```
dx% set slb group <name | all> healthcheck smtp disabled
```

Interval

Interval is the time, in seconds, between each health check request.

- Default: 150 seconds
- Minimum: 1 second
- Maximum: 3600 seconds

Set with the command:

```
dx% set cluster <name> health interval <seconds>
```

Retry

Retry is the number of target host health check requests that must fail before being taken out of rotation.

- Default: 4
- Minimum: 1
- Maximum: 20

Set with the command:

```
dx% set cluster <name> health retry <1-20>
```

Resume

Resume is the number of successful health check requests a down target host must complete before being put back in rotation.

- Default: 1
- Minimum: 1
- Maximum: 20

Set with the command:

```
dx% set cluster <name> health resume <1-20>
```

URL Path

The “URL Path” is the path to the object that the DX appliance should request from the target host, and should not include the site's domain name.

- CORRECT: /products/index.html
- INCORRECT: /products/index.html
- INCORRECT: http://www.juniper.net/products/index.html

Set with the command:

```
dx% set cluster <name> health urlpath </path/to/file.html>
```

Return Code

The “Return Code” is the HTTP response status code that marks a successful health check request.

- The default is 200

Set with the command:

```
dx% set cluster <name> health returncode <HTTP status code>
```

If you are running a web server with NTLM enabled, such as IIS, you will need to change the default health check code from 200 to 401 ("Access Denied"). Otherwise, the cluster target hosts will be marked as L7 down, and the cluster will not work.

Note the following steps:

```
dx% set cluster 3 health returncode 200
(*) dx% write
Writing configuration.
Done.

dx% show cluster 3 target status
TargetHosts:
  [1] 10.0.22.22:80 Layer 7 Down; Unknown Reason
    Total:006 In Use:000 Hot:006 Cold:000 Discards:000

dx% set cluster 3 health returncode 401
(*) dx% write
Writing configuration.
Done.

dx% show cluster 3 health returncode
Health Check Return Code: 401
dx% show cluster 3 target status
TargetHosts:
  [1] 10.0.22.22:80 Up
    Total:008 In Use:000 Hot:006 Cold:002 Discards:000
```

This first shows the user setting the health returncode to 200 (the default value) for a cluster with an NTLM-enabled server in it. Note that the status of the cluster's target status shows that the target server is down. When the health return code is set to 401 ("Access Denied"), the cluster's target status then shows that the target server is up.

Size (optional)

Size is the exact size in bytes of the body of the response for a successful request, as would be reflected in the HTTP Content-Length header. For a web page, this does not include embedded objects such as GIF or JPEG graphics, style sheets, javascript files, etc.

A value of -1 means that the size check is off.

NOTE: This setting is optional and will only be used if a value is provided.

Set with the command:

```
dx% set cluster <name> health size <size in bytes>
```

Clear with the command:

```
dx% clear cluster <name> health size
```

String (optional)

String is the literal, case-sensitive string that must appear in the content of a successful request.

Note that this setting is optional and will only be used if a value is provided, and only applies if the object requested is one of the following MIME types:

- text/html
- text/css
- text/plain
- text/xml
- application/x-javascript

CAUTION: Strings with white space must be enclosed in double quotes.

Set with this command:

```
dx% set cluster <name> health string <string>
```

Clear with this command:

```
dx% clear cluster <name> health string
```

Enabling L7 Health Checking for a Cluster

1. Set the path to the content the DX appliance should use to gauge the health of servers in this cluster:

```
dx% set cluster <name> health urlpath </path/to/file.html>
```

2. Change defaults and enter values for optional settings as previously described.

3. Enable health checking for this cluster:

```
dx% set cluster <name> health enabled
```

4. Save and activate the changes:

```
dx% write
```

Getting Target Host Status Information

To view target host status from the DXSHELL command line, use the command:

```
dx% show cluster <name> health status
```

This is an example of health check output. For information re: interpreting the output, refer to the *Command Line Reference* manual.

```
Health Check Status: enabled
TargetHosts:
   [1] 66.218.71.87:80   Up
       Total:003 In Use:000 Hot:003 Cold:000 Discards:000
   [2] 66.218.71.88:80   Layer 7 Down; Pending Change to Up
       Total:003 In Use:001 Hot:002 Cold:000 Discards:000
   [3] 66.218.71.89:80   Layer 7 Down; Return Code Mismatch
       Total:003 In Use:000 Hot:002 Cold:001 Discards:000
   [4] 66.218.71.90:80   TCP Layer Down; Unknown Reason
       Total:003 In Use:000 Hot:003 Cold:000 Discards:000
```

You can have the DX appliance E-mail you an ALERT when a target host goes down. For additional information, refer to the event notification example "Receive Notification of L7 Health Check Errors via E-Mail" in "Common Administration Tasks" on page 45.

Layer 7 Health Logging System Log Messages

After Layer 7 health checking is enabled on a cluster, the system log will record messages for "Contacting the Target", "Passed L7 Health Check", or "Failed L7 Health Check". These system log messages are generated by default as soon as a user enables L7 health checking on a target. An example of the messages generated:

```
[2005-02-07 15:03:22 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 15:03:17 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 14:28:28 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
failed layer 7 health check.
]
[2005-02-07 14:28:23 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
failed layer 7 health check.
]
[2005-02-07 14:26:32 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 14:26:27 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 14:13:19 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
failed layer 7 health check.
]
[2005-02-07 14:13:14 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
failed layer 7 health check.
]
[2005-02-07 14:12:28 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 14:12:23 (+0800)][ALERT][LSTN][VIP 10.0.21.181:80 Up.]
```

```
[2005-02-07 14:12:22 (+0800)][ALERT][LYR7][Target Server 10.0.81.20:80
passed layer 7 health check.
]
[2005-02-07 14:12:19 (+0800)][ALERT][TSSN][Target Server 10.0.81.30:80 has
been contacted.]
[2005-02-07 14:12:14 (+0800)][ALERT][TSSN][Target Server 10.0.81.20:80 has
been contacted.]
[2005-02-07 14:12:09 (+0800)][ALERT][LYR7][Target Server 10.0.81.30:80
passed layer 7 health check.
]
[2005-02-07 14:12:04 (+0800)][ALERT][LSTN][VIP 10.0.21.181:80 Up.]
[2005-02-07 14:11:59 (+0800)][ALERT][LYR7][Target Server 10.0.81.20:80
passed layer 7 health check.
]
[2005-02-07 14:11:54 (+0800)][ALERT][TSSN][Target Server 10.0.81.30:80 has
been contacted.]
[2005-02-07 14:11:49 (+0800)][ALERT][TSSN][Target Server 10.0.81.20:80 has
been contacted.]
[2005-02-07 14:11:44 (+0800)][ALERT][LSTN][VIP 10.0.21.181:80 Up.]
[2005-02-07 14:11:39 (+0800)][ALERT][LYR7][Target Server 10.0.81.20:80
passed layer 7 health check.
```

Notes on Layer 7 Health Checking

The DX appliance assumes all target hosts are down when L7 health checking is turned-on, and only logs state transitions. This means that with two servers to be checked when we turn-on L7 health checking (one down and one up), the server that is up will be logged in the system log as “Server A passed L7 Health Check” but the server that is down will never be mentioned in the logs until such time as it comes up.

For example:

- Server 0.0.31.20 is normal. It responds to both a ping and an HTTP request (machine is up, the web server is up).

- Server 10.0.31.10 is in a semi-bad state. It responds to a ping, but not an HTTP request (machine is up, the web server is down)

In this state, when L7 health checking is first enabled, you will never see 10.0.31.10 marked as “bad” by L7 health checking. This is because it was never seen as “up” by the DX appliance, and therefore, there was never a transition to record.

Using your SLB's Layer 7 Health Checking

In most situations, it is more appropriate to use the built-in DX appliance L7 health check, however, it is still possible to have your SLB perform L7 health checking. If you want to use your SLB to perform L7, content-based health check on your target web servers, you must assign each target host to its own cluster on the DX appliance.

One-to-one Cluster to Server Mapping

Typically, you would create a single cluster that contains a group of target web servers as:

```
Cluster VIP      Target Hosts
-----------      --------------
1.2.3.4:80   ->  target A, target B, target C, target D
```

However, this configuration will break SLB L7 health checking because the four target web servers appear as a single server to the SLB. If the SLB detects an error, it has no way of knowing which server is down and would mark the whole cluster as down.

To use SLB L7 health checking, create a cluster for each target host on the DX appliance. Note that each cluster requires a distinct IP port combination.

```
Cluster VIP      Target Hosts
-----------      --------------
1.2.3.4:80   ->  target A
1.2.3.5:80   ->  target B
1.2.3.6:80   ->  target C
1.2.3.7:80   ->  target D
```

Conserving IPs with One-to-One Mapping

If your web servers use public addresses or you need to conserve IPs for some other reason, you can still use the one-to-one mapping as previously described. Instead of using a unique IP address for each cluster's VIP, you can give each cluster the same IP with a unique port.

```
Cluster VIP      Target Hosts
-----------      --------------
1.2.3.4:80   ->  target A
1.2.3.4:81   ->  target B
1.2.3.4:82   ->  target C
1.2.3.4:83   ->  target D
```

Scriptable Health Checking

Scriptable Health Checking allows you to write Expect/Tcl scripts that can dynamically pause and unpause target hosts. For example, a script can be written to do an “HTTP GET” on a particular target host. If the HTTP result code is unexpected, the target host can be taken out of rotation. You import the script into the DX appliance, configure it for execution, and execute it.

Scriptable Health Checking requires a license from Juniper Networks before it can be used. Contact your Juniper Networks Sales Representative for information.

Expect/Tcl Scripts

Capturing and Configuring Expect/Tcl Scripts

You import the Expect/Tcl scripts using the Command Line Interface. Once the script is imported, the DX appliance validates it by checking for syntax errors. The maximum size of a script is 1 MByte, and there is no restriction on the total number of scripts.

NOTE: The DX appliance checks the script for correct syntax only, not for proper operation. It is possible to write a script that is syntactically correct, but that will produce errors or unexpected results during operation. Use discretion when coding Expect/Tcl scripts.

The scripts can be configured to run once or execute at an interval. The DX appliance allows you to delete Expect scripts that are not configured. You cannot edit expect scripts on the DX appliance. There must be a minimum of 1MByte of free disk space for the capture to commence.

Run-Time Environment

The DX Application Acceleration Platform will not allow the script to damage or delete software running on the DX appliance. The script may, however, purge its sandbox environment. If this does happen, new scripts may not run as designed until the sandbox environment is repaired.

The DX appliance reports runtime script errors back to the user in the logs. The scripts are able to hard-pause, soft-pause, and un-pause hosts. The available hosts are target hosts of the type:

- HTTP and forwarder clusters
- SLB

Pausing and unpausing changes that are script generated are written to memory only, and will be lost across reboots.

Running scripts are killed:

- Upon a related configuration change (new IP address, port change, health configuration updated, etc.).

NOTE: This means that any scripts that are running will be killed and restarted based on the new configuration. This is captured as "Forced Termination" in the script statistics.

- When a scheduled script is executed by the DX appliance (non-test mode), any existing scripts of the same name will be terminated.
- When writing a script, the script must use the following path as the first line:

```
#!/usr/bin/expect -f
```

Sandbox Environment

Network communication is through an IP address assigned to the sandbox. The DX appliance does not allow file writes that are script invoked.

The following resource limitations apply when a script is executing:

- Total script size memory limit is 5 MB.
- Health scripts run at a lower priority than server processes to ensure that script doesn't take up CPU time when server processes are running.
- Only 32 pseudo-terminals are available; ptys(32).

This affects the expect command "`spawn`" which uses pseudoterminals (ptys) to launch the corresponding process. Since these are limited, this effectively means health scripts cannot run the spawn command more than 32 times. However, the same ptys are also used by other portions of the system (such as the Command Line Interface), so in practice this number is much less. In general scripts should minimize the amount of time they hold on to the pty to avoid this scenario and avoid launching many processes using the "spawn" command.

These binaries are provided in the sandbox:

- ping <host>
- ssh
- telnet
- openssl
- nslookup
- traceroute

For the ping command, the only command line argument allowed is <host>. For the remainder of the commands, all standard command line options are allowed.

Scriptable Health Checking Tcl API

The API commands used with Scriptable Health Checking are:

```
rln_send_event -i ip -p port -e event -c class -m msg
```

where:

- “ip” is the address of the target host (required field). The IP address can be specified in either traditional dotted format (192.168.0.80) or in hexadecimal format (0xC0A80050).
- “port” is the port number of the target host. Default value is 0. A value of 0 means all ports.

NOTE: When the port number is available, the port should be explicitly specified. This will result in better performance.

- “event” is the event to be generated. Valid values are up or down (required fields).
- “class” is the class of the event. Valid values are layer7, layer5, layer4, layer3, or none. Default class is none.
- “message” is the log message accompanying the event. The default message is empty.

```
rln_send_action -i ip -p port -a action -m msg
```

where:

- “ip” is the address of the target host (required field). The IP address can be specified in either traditional dotted format (192.168.0.80) or in hexadecimal format (0xC0A80050).
- “port” is the port number of target host. Default value is 0. A value of 0 means all ports.
- “action” is the suggested action. Values are hard-pause, soft-pause, or un-pause (required fields).
- “message” is the message accompanying the event. The default message is empty.

```
rln_send_log -l location -m msg
```

where:

- “location” decides the log destination. Only “health check” is supported as of this release. The default value is `healthcheck`.
- “message” is the message accompanying the event. The default message is empty.

```
rln_radius_auth -i ip -p port -k serverkey -t timeout -r retries -u username -w password
```

where:

- "ip" is the ip address of the RADIUS server.
- "port" is the port of the RADIUS server.
- "serverkey" is the client's RADIUS secret. The RADIUS server has a secret for every client ip.
- "timeout" is the time in seconds after sending a RADIUS request that the client waits for a response. If the response is not received within <timeout> seconds, the RADIUS request is resent.
- "retries" is the number of times the client resends the RADIUS request for a response before determining the server as down.
- "username" is the name of the user to be authenticated.
- "password" is the password of the user to be authenticated.

```
rln_ldap_auth -i ip -p port -d admin_user_dn -s admin_password -b base_dn
-a user_attribute -v version -c ca_cert_file -u username -w password
```

where:

- "ip" is the ip address of the LDAP server.
- "port" is the port of the LDAP server.
- "admin_user_dn" is the DN (Distinguished Name) of the admin user. This field is optional. When present, admin authentication is done before user authentication. When absent, admin authentication is not done and user authentication is done directly.
- "admin_password" is the password for the admin user. This field is optional, but it must be present when "admin_user_dn" is present.
- "base_dn" is the DN of the root of the tree in the LDAP database under which the LDAP search has to be done for the users.
- "user_attribute" is the name of the attribute uniquely identifying the users in the LDAP database.
- "version" is the LDAP version to be used.
- "ca_cert_file" is the name of the file containing trusted root ca certificates. This field is optional. When present, LDAP connection with the server is upgraded to a TLS connection before doing the authentication. When absent, LDAP connection with the server is in clear text. This field has to be provided to use LDAP over TLS.
- "username" is the name of the user to be authenticated.
- "password" is the password of the user to be authenticated.

The Expect/Tcl Command Set

The Expect/Tcl commands are used to update target host status and send log messages. The DX appliance has a subset of the TCL command set. This subset has only commands that are deemed safe and are needed for script writing. Commands such as `fork`, `exec`, and `filewrites` have been removed.

Available Tcl/Expect Commands

Table 8 shows the currently available TCL commands.

Table 8: TCL Commands

Command	Command	Command	Command	Command
Safe Base	eval	interp	proc	tcl_startOfPreviousWord
Tcl	exit	join	puts	tcl_wordBreakAfter
after	expr	lappend	re_syntax	tcl_wordBreakBefore
append	fblocked	lindex	read	tcltest
array	fconfigure	linsert	regexp	tclvars
auto_qualify	fileevent	list	registry	time
bgerror	filename	llength	regsub	trace
binary	flush	lrange	resource	udp_conf
break	for	lreplace	return	udp_peer
catch	foreach	lsearch	scan	unknown
clock	format	lset	set	unset
close	gets	lsort	socket	uplevel
concat	global	memory	split	upvar
continue	history	msgcat	string	variable
dde	http	package	subst	vwait
encoding	if	parray	switch	while
eof	incr	pkg::create	tcl_endOfWord	
error	info	pkg_mkIndex	tcl_startOfNextWord	

Table 9 shows the expect commands that are supported

Table 9: Supported Expect Commands

Command	Command	Command	Command	Command
close	expect_after	match_max	send_error	timestamp
exit	expect_background	overlay	send_tty	trap
exp_continue	expect_before	parity	send_user	wait
exp_open	expect_tty	promp1	sleep	
exp_pid	expect_user	prompt2	spawn	
exp_version	getpid	remove_nulls	strace	
expect	log_user	send	stty	

Table 10 shows the expect commands that are NOT supported.

Table 10: Expect Commands that are Not Supported

Command	Command	Command	Command	Command
debug	exp_internal	inter_return	interpreter	send_log
disconnect	fork	interact	log_file	system

Logging and Statistics

The script generates information-level logs that are logged in a new health script log. ALERT system logs are generated for various failures. Some sample scenarios are:

- Cannot kill old scripts
- Cannot initialize the configuration
- Cannot setup for script launching
- Memory error
- Script terminates abnormally
- Periodic script is killed due to the next run interval
- Script configuration error
- Script launch error

Statistics are provided to report the state of each script. The following data points are available:

- Is the script running?
- The number of times a script has been launched.
- The number of times the script failed to start.

- The number of times a script failed after it started.
- The number of times script killed due forced termination (configuration change or the next script interval due).
- The number of successful runs.
- The last run (in UTC).
- The next run (in UTC).

Statistics can be cleared using the DXSHELL.

If you see too many “Force Termination” failures for a periodic script, it could mean that the periodic interval is too short. The script is not finishing in time, and is killed and restarted for the next run. Configuration changes will also kill currently running scripts that would also increment this statistic.

The statistics are only updated/captured for scripts automatically run by a scriptable health system. They are not updated for scripts that are run manually from the DXSHELL using the command `set health script <script_name> testrun`. This command was created so that operators can perform a test run of the script before adding the script for automatic running via the health system.

When the script is run manually, the operator can visually see whether or not the script ran successfully, so the statistics are not updated. The status for scripts that run automatically cannot be seen directly by the operator. Instead, the operator must query the statistics to see the status information.

TCL UDP Extension

The TCL User Datagram Protocol extension (known as tcludp) provides commands to create and use a UDP socket. To use the extension, the script has to load the UDP package by adding “package require udp” in the tcl/expect file. Some useful UDP commands that are supported are:

```
udp_open [port]
```

`udp_open` will open a UDP socket. If port is specified the UDP socket will be opened on that port. Otherwise the system will choose a port and the user can use the `udp_conf` command to obtain the port number if required.

```
udp_conf sock host port
```

`udp_conf` in this configuration is used to specify the remote destination for packets written to this sock. You must call this command before writing data to the UDP socket.

```
udp_conf sock [-myport] [-remote] [-peer] [-broadcast bool] [-ttl count]
```

In addition to being used to configure the remote host, the `udp_conf` command is used to obtain information about the UDP socket.

- “myport” returns the local port number of the socket.

- “remote” returns the remote hostname and port number as set using the udp_conf sock host port.
- “peer” returns the remote hostname and port number for the packet most recently received by this socket.
- “broadcast [boolean]” UDP packets can listen and send on the broadcast address. For some systems, a flag must be set on the socket to use broadcast. With no argument, this option will return the broadcast setting. With a boolean argument, the setting can be modified.
- “ttl [count]” The time-to-live is given as the number of router hops the packet may do. For multicast packets this is important in specifying the distribution of the packet. The system default for multicast is 1 which restricts the packet to the local subnet. To permit packets to pass routers, you must increase the ttl. A value of 31 should keep it within a site, while 255 is global.

```
udp_conf [-mcastadd groupaddr]
udp_conf [-mcastdrop groupaddr]
```

tcludp sockets can support IPv4 multicast operations. To receive multicast packets the application has to notify the operating system that it should join a particular multicast group. These are specified as addresses in the range 224.0.0.0 to 239.255.255.255.

```
udp_peek sock [buffersize]
```

Examines a packet without removing it from the buffer. This function is not available on windows.

Command line arguments enclosed in [square brackets] are optional.

Scriptable Health Checking Commands

Use these commands to configure scriptable health checking:

Configuration Commands

To capture a script, type the command:

```
dx% import health script <scp or tftp path>
```

The maximum script name length is 64 characters.

To add the script, type the command:

```
dx% add health script <script_name>
```

To enable or disable the script, type the command:

```
dx% set health script <script_name> <enabled | disabled>
```

To set the script vip, type the command:

```
dx% set health script <script_name> vip <vip>
```

For this command, the DX appliance determines the most appropriate interface to alias the IP address.

To set the script execute interval, type the command:

```
dx% set health script <script_name> interval <value>
```

If a zero is set, the script will only run once. A value greater than zero specifies the run interval in seconds. The maximum value is 86400 seconds.

To perform a test run of the health script, type the command:

```
dx% set health script <script_name> testrun
```

This command allows you to test drive a health script and visually inspect the results to see if the script is behaving properly. You can put debug messages to trace your logic and check the health logs to see if the health check status is being communicated properly by the script. When the script finishes (it might not finish if it's a run once script), you can see the exit status to see if it ran successfully. Once you are comfortable with this, you can enable the script for automatic execution by the scriptable health system.

To delete a health script configuration node, type the command:

```
dx% delete health script <script_name>
```

To delete a script file, type the command:

```
dx% delete file <script_name>
```

Show Commands

To show the configuration of Scriptable Health Checking, use the commands:

```
dx% show health script <script_name | all> interval
dx% show health script <script_name | all> vip
dx% show health script <script_name | all> name
dx% show health script <script_name | all> status
dx% show health script <script_name | all> stats
```

To clear the Scriptable Health Checking statistics, type the command:

```
dx% clear health script <script_name | all> stats
```

Logging Commands

To show the health script log, type the command:

```
dx% show log health script
```

To clear the health script log, type the command:

```
dx% clear log health script
```

To export the health script log, type the command:

```
dx% export log health script <destination>
```

Capture and Configuration Example

This is an example of how to capture and configure a script:

```
dx% import health script tftp://qa/scripts/foo.exp
done. 255 bytes transferred.
dx% add healthscript foo.exp
added healthscript foo.exp.
(*) dx% set health script foo.exp vip 192.168.14.75
(*) dx% set health script foo.exp interval 10
(*) dx% set health script foo.exp enabled
(*) dx% write
Writing configuration.
Done.
```

Sample Scripts

This sample script sends a L7 down event:

```
#!/usr/bin/expect -f

package require RlnTclExt
set thost 192.168.14.221
set port 80
if { $argv == “debug” }     {
    set dbgflag -d
}

#Host $thost is down.
rln_send_event -i $thost -p $port -e down -c layer7 -m “http GET failed”
This sample script sends a hardpause message:
#!/usr/bin/expect -f

package require RlnTclExt
set thost 192.168.14.221
set port 80
if { $argv == “debug” }     {
    set dbgflag -d
}

#Host $thost is down.
rln_send_action -i $thost -p $port -a hardpause
```

Chapter 10

Setting up the DX Appliance for “Sticky” Traffic

This chapter describes setting up the DX appliance for sticky traffic on the DX Application Acceleration Platform, discussing the following topics:

Overview

“Sticky” is the common term used to describe web client requests being redirected to the same target host within a cluster. The client “sticks” with the server.

To configure the DX appliance to create sticky connections between clients and target servers, simply specify on a cluster-by-cluster basis whether you want the cluster to use cookie or client-IP based stickiness, and then set a timeout value for the sticky connection.

To disable client stickiness, use the following commands:

```
dx% set cluster <name> sticky method none
```

The none option is the default setting on the DX appliance.

Configuration Instructions for Cookie-Based Client Stickiness

To choose cookie-based stickiness, use the following commands:

```
dx% set cluster <name> sticky method cookie
dx% set cluster <name> sticky cookie expire [0-3000000]
```

The allowable range of cookie expire values is 1 minute to 3,000,000 minutes (5.71 years). Setting the cookie expire value to 0 means that the cookies never expire.

To display the sticky settings for this cluster:

```
dx% show cluster <name> sticky
```

To disable cookie-based client stickiness, set the sticky method to none.

```
dx% set cluster <name> sticky method none
```

Sticky cookies have been known to break some server applications. An option has been added to remove the sticky cookie from the request headers based upon the configuration. When there are multiple cookies in a “Single Cookie” header, it only strips the sticky cookie.

To control whether the cookie is stripped, type the command:

```
dx% set cluster <name> sticky passheader [disabled | enabled*]
```

When enabled, the cookie is passed through; when disabled, the cookie is stripped:

Configuration Instructions for Client IP-Based Stickiness

To choose client IP-based stickiness, use the following commands:

```
dx% set cluster <name> sticky method clientip
dx% set cluster <name> sticky clientip timeout [1-43200]
```

The range of timeout values is 1 minute to 43200 minutes (30 days).

You can select the appropriate method for hashing depending upon whether the DX appliance is deployed in front of a public web site or in front of an intranet site. The command used to set this parameter is:

```
dx% set cluster <name> sticky clientip distribution <internet | intranet>
```

For optimum performance for a public web site, set to "internet" using the command:

```
dx% set cluster <name> sticky clientip distribution internet
```

For optimum performance for an intranet web site, set to "intranet" using the command:

```
dx% set cluster <name> sticky clientip distribution intranet
```

To display the sticky settings for this cluster:

```
dx% show cluster <name> sticky
```

To disable clientIP-based client stickiness, set the sticky method to none:

```
dx% set cluster <name> sticky method none
```

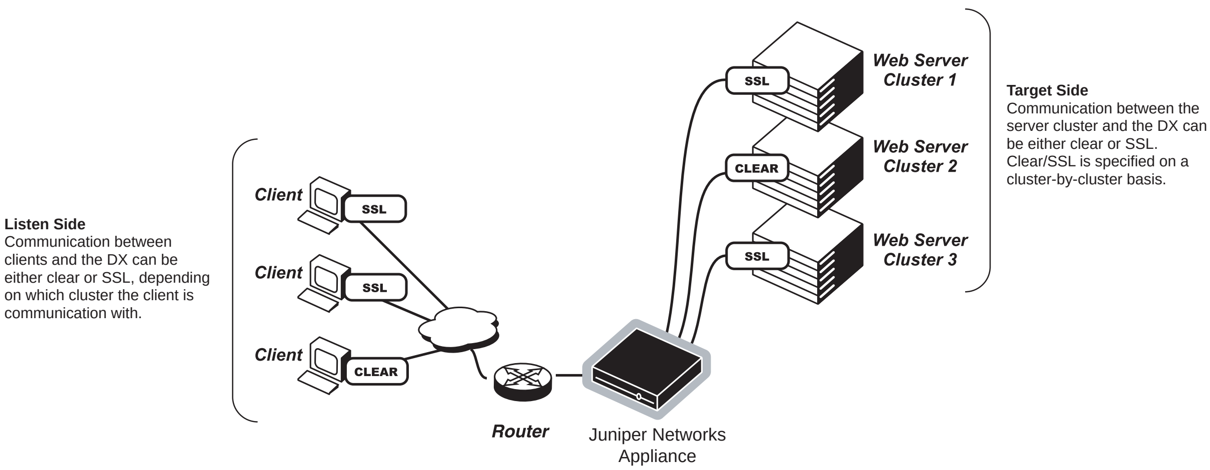
Chapter 11
Setting Up the DX Appliance for SSL Traffic

This chapter describes setting up the DX appliance for Secure Socket Layer (SSL) traffic, discussing the following topics:

Overview

Configuring the DX appliance to serve data using SSL is easy. The DX appliance sits in front of your server(s), holds your site certificates and keys, and processes the incoming and outgoing SSL transactions. This off-loads resource-intensive SSL processing from your servers and allows them to focus on serving content. Refer to Figure 40.

Figure 40: Listen and Target-Side Illustration



The DX appliance can also act as an SSL Forwarder. In Forwarder mode, the DX appliance performs the SSL encryption or decryption, and then forwards the HTTP or non-HTTP traffic directly to the server or client. In the Forwarder mode, the client connection gets terminated at the DX appliance, and the DX appliance opens a new connection to the server. The DX appliance then forwards HTTP and non-HTTP traffic transparently from the client to the server, which means it never initiates termination of a connection. That is done by either the client or the server.

This chapter will explain the basic conventions, terms and commands used by the when working with SSL traffic, including step-by-step instructions for various SSL configurations. It assumes that you already have valid certificates and keys for use in an actual production environment. For testing purposes, the DX appliance comes with mock certificate and key files named, respectively, democert and demokey. Additional certificates and keys can be generated by the DX appliance for testing purposes (refer to “GEN KEY” on page 183 for additional information).

Before You Begin

If you are installing the DX appliance in a testing environment where valid key and certificate files are not needed, the DX appliance comes with “dummy” key and certificate files named demokey and democert, respectively.

If you are installing the DX appliance in a production environment, make sure you have valid key and certificate files in base-64 encoded format. Instructions for importing these files from a variety of environments, as well as converting them to base-64, appear in “Importing Existing Keys and Certificates” on page 174.

When importing key files from different environments, occasionally they will need to be converted using the OpenSSL software. For information on this program, see the openSSL web pages at:

http://www.openssl.org/

Basic Conventions and Terms

Data travels between the server cluster and the DX appliance, and also between the DX appliance and the client browser. Data that flows between the DX appliance and the client browser is termed “Listen” traffic. Data that flows between the DX appliance and the target server cluster is termed “Target” traffic.

- LISTEN Traffic: is traffic between the DX appliance and the client browser
- TARGET: Traffic: is traffic between the DX appliance and the server the DX appliance is accelerating
- SSL settings for the target and listen sides are set independently

Whether the DX appliance uses SSL is specified on a cluster-by-cluster basis. For example, for cluster 1 the DX appliance can have SSL enabled on the listen side and disabled on the target side, while for cluster 2 the DX appliance can have SSL enabled on both sides, etc.

With these two major divisions in mind, let's look at an already-configured server cluster named cluster 1 (`dx%` represents the command prompt).

```
dx% show cluster 1
Cluster [1]
Description:
Listen Address: 1.1.1.1
Listen Netmask: 255.255.255.255
Listen Port: 100
Listen SSL Status: disabled
Listen SSL Protocol: sslv23
Listen SSL Certfile: testtest2_selfcert
Listen SSL Keyfile: testtest2
Listen SSL Keypass: none
Listen SSL Ephemeral Keyfile:
Listen SSL Ephemeral Keypass: none
Listen SSL Ciphersuite: all
Listen SSL Cipherfile:
Client Authentication: disabled
CA Certfile:
```

```
CA CRL File:
CA Trust File:
Client Certificate Authentication Type: local
Client Certificate Forwarding: disabled
Client Certificate Forwarding Format: DERBase64
Listen TargetsDown Mode: blackhole
DSR Status: disabled
Health Check Status: disabled
Health Check Interval: 150
Health Check Retry: 4
Health Check Resume: 1
Health Check Url Path:
Health Check Return Code: 200
Health Check Size: -1
Health Check String:
Health Check Timeout: 15
Health Check User Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0;
T312461)
Sticky Method: none
Sticky Cookie Mask: ipport
Sticky Cookie Expire: 0
Sticky Client IP Distribution: internet
Sticky Client IP Timeout: 120
Convert 302 Protocol Status: disabled
Weblog Status: disabled
Weblog Destination: syslog
Weblog Format: common
Weblog Syslog Host:
Weblog Syslog Port: 514
Weblog Batch memory allocated for this cluster (in MB): 10
Total free memory available for all clusters: 50 MB
Weblog Batch Copy Time 1:
Weblog Batch Copy Time 2:
Weblog Batch Copy Time 3:
Weblog Batch Retry Interval: 60
Weblog Batch Scp Directory:
Weblog Batch Scp Username:
Weblog Batch Scp Keyfile:
Weblog Batch Host:
Weblog Batch Compression: enabled
Weblog Delimiter: space
Connection Binding Status: disabled
Rule Set File:
AppRule Processing: disabled
AppRule Limit Retry Post: 32768
HTTP Authentication Status: disabled
HTTP Authentication Method: WWW
Authentication Realm:
Authentication Response Text:
HTTP Authentication Protocol: RADIUS
Authentication Redirect Status: disabled
Authentication Redirect Page URL: /auth.shtml
Authentication Redirect Host:
Authentication Redirect Protocol: http
Authentication Password MaxAge: 1
Authentication Password MaxLength: 8
HTTP Authentication Cache Status: enabled
Authentication Cache MaxAge: 60
RADIUS Server Key:
RADIUS Server Timeout: 10
RADIUS Server Retries: 3
RADIUS Server 1 IP:
RADIUS Server 1 Port: 1812
```

```
RADIUS Server 2 IP:
RADIUS Server 2 Port: 1812
HTTP Authentication Auditing: enabled
Audit Level: failures
OWA Status: disabled
Client IP Transparency: disabled
Targetname: www1.yourdomain.com
Target SSL Status: disabled
Target SSL Protocol: sslv23
Target SSL Certfile:
Target SSL Keyfile:
Target SSL Keypass: none
Target SSL Ciphersuite: common
Target SSL Cipherfile:
Target SSL Timeout: 1440
Target Local IP:
TargetHosts: none
Cache: None.
dx%
```

This is what all the parameters that pertain to SSL mean:

- Listen Port: 443

  443 is the standard port for SSL traffic. This is the port through which the DX appliance communicates with client browsers.

- Listen SSL Status: Enabled

  This line specifies whether communication between the DX appliance and client browsers will use SSL or not. A value of “enabled” means that SSL will be used. A value of “disabled” means SSL will not be used.

NOTE: If the value is set to “disabled,” the next five values will be ignored (Listen SSL Protocol, Listen SSL Certfile, Listen SSL Keyfile, Listen SSL Keypass, and Listen SSL Ciphersuite).

- Listen SSL Protocol: sslv23

  DX appliance supports several protocols for communicating securely with client browsers: sslv2, sslv3, sslv23, and tlsv1. Typically you will choose sslv23 for listen traffic to allow communication with the greatest number of SSL-supported client browsers.

- Listen SSL Certfile: democert

  In this case, democert is the sample certificate shipped with the DX appliance. This is where you would enter the name of your actual certificate file.

NOTE: Democert must be used with demokey.

- Listen SSL Keyfile: demokey

  As previously described, demokey is the sample key shipped with the DX appliance. This is where you would enter the name of your actual keyfile.

NOTE: demokey must be used with democert.

- Listen SSL Keypass: none

  If your private key is encrypted with a password, enter that password here. If a password has been entered, then ******** is displayed. Otherwise, none is displayed.

- Listen SSL Ciphersuite: all

  Here you have the following options: strong, export, common, and all. For an explanation of each cipher suite and a list of included ciphers, refer to the section "SSL Cipher Suite Details" at the end of this chapter. (Generally for listen traffic you will use the all ciphersuite, which will allow communication with the widest number of SSL-supporting client browsers.)

- Targetname: www1.yourdomain.com

  This is the target name of cluster 1.

- Target SSL Status: Disabled

  This line specifies whether communication between the DX appliance and cluster 1 will use SSL or not. A value of "disabled" means SSL will not be used. A value of "enabled" means that SSL will be used.

NOTE: If the value is set to "disabled," the next six values will be ignored (Target SSL Protocol, Target SSL Certfile, Target SSL Keyfile, Target SSL Keypass, Target SSL Ciphersuite, and Target SSL Timeout).

- Target SSL Protocol: sslv23

  This is where you specify the SSL protocol with which DX appliance and cluster 1 will communicate. DX appliance supports several protocols for communicating securely with target clusters: sslv2, sslv3, sslv23, and tlsv1.

NOTE: It is possible to do target-side SSL without the following three fields. It will result in SSL without client authentication.

- Target SSL Certfile

  If the DX appliance is communicating via SSL with the target cluster, this is where you would enter the filename of the cluster's certificate. If SSL between the DX appliance and the target cluster is disabled, you do not need to enter a filename here. Note that it is only used for client authentication.

- Target SSL Keyfile

  If the DX appliance is communicating via SSL with the target cluster, this is where you would enter the name of the cluster's keyfile. If SSL between the DX appliance and the target cluster is disabled, you do not need to enter a filename here. Note that it is only used for client authentication.

- Target SSL Keypass: none

  If cluster 1's private key is encrypted with a password, enter that password here. Note that it is only used for client authentication.

- Target SSL Ciphersuite: all

  Here you have the following options: strong, export, common, and all.

- Target SSL Timeout: 1440

  This allows the DX appliance to timeout the SSL session with the target cluster. The value is in minutes.

  ```
  TargetHosts:
   [1] 192.168.0.157:80 (enabled)
   [2] 10.0.11.81:80 (disabled)
  ```
  These are the servers that the DX appliance is accelerating.

Step-by-step Configuration Examples

NOTE: An (*) before the command prompt indicates that the configuration has been changed but not written (SAVED).

Possible SSL Cluster Configurations with the DX Appliance

There are four possible SSL Cluster Configurations. Each is discussed in an example that follows.

LISTEN: SSL Disabled
TARGET: SSL Enabled

LISTEN: SSL Enabled
TARGET: SSL Disabled

LISTEN: SSL Enabled
TARGET: SSL Enabled

LISTEN: SSL Disabled
TARGET: SSL Disabled

SSL Configuration Examples: Listen: Enabled and Target: Disabled

These instructions will guide you through the process of setting up a DX appliance with SSL "enabled" on the listen side and "disabled" on the target side. This section assumes you have already captured your key and certificate files. Refer to the line-by-line explanations of these commands in "Basic Conventions and Terms" on page 161.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 443
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% write
dx% set cluster 1 listen ssl protocol sslv23
(*) dx% set cluster 1 listen ssl certfile cert
(*) dx% set cluster 1 listen ssl keyfile key
(*) dx% set cluster 1 listen ssl keypass
New password:
(*) dx% set cluster 1 listen ssl ciphersuite all
(*) dx% set cluster 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% set cluster 1 target host all enabled
(*) dx% set cluster 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

4. Enable the convert302 protocol option.

   With the convert302 protocol option enabled, the DX appliance will convert the HTTP 302 responses from the target server from HTTP to HTTPS for the client.

   ```
   dx% set cluster <name> convert302protocol enabled
   ```

NOTE: If you need to redirect requests from a secure server back to the non secure server, you should not enable this option.

You should now have SSL on the listen side and clear on the target side. Try opening a browser and going to https://10.100.2.63/ to test the configuration.

SSL Configuration Examples: Listen: Disabled and Target: Enabled

These instructions will guide you through the process of setting up a DX appliance with SSL "disabled" on the listen side and "enabled" on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

   ```
   dx% set cluster 1 listen port 80
   (*) dx% set cluster 1 listen vip 10.100.2.63
   (*) dx% set cluster 1 listen ssl disabled
   (*) dx% write
   ```

2. Set the target configuration:

   ```
   dx% set cluster 1 target name mywebserver.juniper.net
   (*) dx% clear cluster 1 target host all
   (*) dx% set cluster 1 target host 10.100.1.37:80
   (*) dx% set cluster 1 target host all enabled
   (*) dx% write
   dx% set cluster 1 target ssl protocol sslv23
   (*) dx% set cluster 1 target ssl ciphersuite all
   (*) dx% set cluster 1 target ssl timeout 1440
   (*) dx% set cluster 1 target ssl enabled
   (*) dx% set server factory cscf enabled
   (*) dx% write
   ```

3. OPTIONAL: If the web server certificates are invalid and being used for testing:

   ```
   dx% set server factory cscf disabled
   (*) dx% write
   ```

4. Start the server:

   ```
   dx% set server up
   (*) dx% write
   ```

You should now have clear on the listen side and SSL on the target side. Open a browser and go to https://10.100.2.63/ to test the configuration.

SSL Configuration Examples: Listen: Enabled and Target: Enabled

These instructions will guide you through the process of setting up a DX appliance with SSL "enabled" on the listen side and "enabled" on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 443
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% write
dx% set cluster 1 listen ssl protocol sslv23
(*) dx% set cluster 1 listen ssl certfile txcert
(*) dx% set cluster 1 listen ssl keyfile txkey
(*) dx% set cluster 1 listen ssl keypass
New password:
(*) dx% set cluster 1 listen ssl ciphersuite all
(*) dx% set cluster 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% set cluster 1 target host all enabled
(*) dx% write
dx% set cluster 1 target ssl protocol sslv23
(*) dx% set cluster 1 target ssl ciphersuite all
(*) dx% set cluster 1 target ssl timeout 1440
(*) dx% set cluster 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the web server certificates are invalid and being used for testing.

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end SSL. Open a browser and go to https://10.100.2.63/ to test the configuration.

SSL Configuration Examples: Listen: Disabled and Target: Disabled

These instructions will guide you through the process of setting up a DX appliance with SSL "disabled" on the listen side and "disabled" on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 80
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% set cluster 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% set cluster 1 target host all enabled
(*) dx% set cluster 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end clear. Try opening a browser and going to http://10.100.2.63/

SSL Forwarder Configuration

The DX appliance can be configured to act as an SSL Forwarder. In Forwarder mode, the DX appliance performs the SSL encryption or decryption, and then forwards the HTTP or non-HTTP traffic directly to the server or client. In Forwarder mode, the client connection gets terminated at the DX appliance, and the DX appliance opens a new connection to the server. The DX appliance then forwards HTTP and non-HTTP traffic transparently from the client to the server. This means that the DX appliance never initiates termination of a connection; it is either the client or the server.

An SSL Forwarder offers these features:

- Forwards HTTP and non-HTTP traffic transparently from client to server
- Forwarder can be used for:
  - Offloading SSL on the client side for HTTP and non-HTTP traffic
  - Server side SSL for HTTP and non-HTTP traffic
  - End-to-end SSL
- Performs Layer 4 health checking for monitoring target hosts
- Provides I/O and SSL statistics (same as a cluster)

- Acts like a cluster with connection-binding “On” and pre-established (“Hot”) target connections equal ‘Zero”, and no HTTP handling.
- Honors all global factory settings applicable to I/O and SSL layers
- Supports DSR mode

Possible SSL Forwarder Configurations with the DX Appliance

There are four possible SSL Forwarder Configurations. Each is discussed in an example that follows:

LISTEN: SSL Disabled TARGET: SSL Enabled	LISTEN: SSL Enabled TARGET: SSL Disabled
LISTEN: SSL Enabled TARGET: SSL Enabled	LISTEN: SSL Disabled TARGET: SSL Disabled

SSL Configuration Examples: Listen: Enabled and Target: Disabled

These instructions will guide you through the process of setting up a DX appliance as a Forwarder with SSL “enabled” on the listen side and “disabled” on the target side. This section assumes you have already captured your key and certificate files. Refer to the line-by-line explanations of these commands in “Basic Conventions and Terms” on page 161.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 443
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% write
dx% set forwarder 1 listen ssl protocol sslv23
(*) dx% set forwarder 1 listen ssl certfile cert
(*) dx% set forwarder 1 listen ssl keyfile key
(*) dx% set forwarder 1 listen ssl keypass
New password:
(*) dx% set forwarder 1 listen ssl ciphersuite all
(*) dx% set forwarder 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target host all enabled
(*) dx% set forwarder 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

4. Enable the convert302 protocol option.

   With the convert302protocol option enabled, the DX appliance will convert the HTTP 302 responses from the target server from HTTP to HTTPS for the client.

```
dx% set forwarder <name> convert302protocol enabled
```

NOTE: If you need to redirect requests from a secure server back to the non-secure server, you should not enable this option.

You should now have SSL on the Listen side and clear on the Target side. Try opening a browser and going to https://10.100.2.63/ to test the configuration.

SSL Configuration Examples: Listen: Disabled and Target: Enabled

These instructions will guide you through the process of setting up a DX appliance as a Forwarder with SSL “disabled” on the listen side and “enabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 80
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% set forwarder 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target host all enabled
(*) dx% write
dx% set forwarder 1 target ssl protocol sslv23
(*) dx% set forwarder 1 target ssl ciphersuite all
(*) dx% set forwarder 1 target ssl timeout 1440
(*) dx% set forwarder 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the web server certificates are invalid and being used for testing:

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have SSL clear on the Listen side and enabled on the Target side. Open a browser and go to https://10.100.2.63/ to test the configuration.

SSL Configuration Example, Listen: Enabled, Target: Enabled

These instructions will guide you through the process of setting up a DX appliance as a Forwarder with SSL "enabled" on the listen side and "enabled" on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 443
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% write
dx% set forwarder 1 listen ssl protocol sslv23
(*) dx% set forwarder 1 listen ssl certfile txcert
(*) dx% set forwarder 1 listen ssl keyfile txkey
(*) dx% set forwarder 1 listen ssl keypass
New password:
(*) dx% set forwarder 1 listen ssl ciphersuite all
(*) dx% set forwarder 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target host all enabled
(*) dx% write
dx% set forwarder 1 target ssl protocol sslv23
(*) dx% set forwarder 1 target ssl ciphersuite all
(*) dx% set forwarder 1 target ssl timeout 1440
(*) dx% set forwarder 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the web server certificates are invalid and being used for testing:

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server

```
dx% set server up
(*) dx% write
```

You should now have end-to-end SSL. Open a browser and go to https://10.100.2.63/ to test the configuration.

SSL Configuration Example, Listen: Disabled, Target: Disabled

These instructions will guide you through the process of setting up a DX appliance as a Forwarder with SSL "disabled" on the listen side and "disabled" on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 80
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% set forwarder 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target host all enabled
(*) dx% set forwarder 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end clear. Try opening a browser and going to `http://10.100.2.63/`.

Importing Existing Keys and Certificates

If you already have certificates and keys, you can transfer them to the DX appliance. This section shows how to import keys and certificates from:

- Apache mod_ssl
- ApacheSSL
- IIS 4.0
- IIS 5.0
- iPlanet

Key and certificate file names cannot contain spaces, and must be compatible with the server operating system. When prompted either to name a key or certificate file or check the name of a key or certificate file, ensure that the names follow these conventions. Keys and certificates must be base-64 encoding to work with the DX appliance.

NOTE: If you are using a global certificate, you will need to install a chain certificate (Intermediate Certificate) so that browsers can trust your certificate. Your Trusted Root Certificate Authority can provide this intermediate certificate.

An “Intermediate Certificate” is a certificate issued by the Trusted Root Certificate Authority. As well as issuing SSL certificates, the Trusted Root CA certificate can be used to create another certificate, which in turn will then be used to issue SSL certificates. Any SSL certificates issued by the Intermediate Certificate inherit the trust of the Trusted Root--effectively creating a certification chain of trust as:

Trusted Root CA > Intermediate > SSL Certificate

Currently, Verisign, Inc. and EBIZID use Intermediate Certificates. Other certificate authorities do not use Intermediate Certificates at this time.

Importing from Apache mod_ssl

The key and certificate locations are listed in the $APACHEROOT/conf/httpd.conf file. The default key is $APACHEROOT/conf/ssl.key/*.key. The default certificate is $APACHEROOT/conf/ssl.crt/*.crt. Make note of these names and locations.

To import these files to the DX appliance, follow this example of copying and pasting the key and certificate files from the locations previously described.

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gTO5MWTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQK
EwlERU1PIE9OTFkxEjAQBgNVBAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE9OTFkwHhcNMDIwMzA1MjM1MzAxWhcN
MDIwMzA2MjM1MzAxWjCBizELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gTO5M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQKEwlERU1PIE9OTFkxEjAQBgNV
BAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE9OTFkwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgL5Z5tcp8
HkubHFrpC1tub2CEANVBJsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAkoO0d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwHQYDVR0OBBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgbAwga2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQQGEwJYWDESMBAGA1UECBMJREVNTyBPTkxZ
MRIwEAYDVQQHEwlERU1PIE9OTFkxEjAQBgNVBAoTCURFTU8gTO5MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRIwEAYDVQQDEwlERU1PIE9OTFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gTO5MWYIBADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
W0FOiR7MsY64y5cbpMoGrfZ2qNgNKF+i6WLlmTfh4+1tKiCMnhTRPMcszjvwgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
.
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQWkhMCb+BrlT7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDJiQdQwOYlmaWQ7x5hSIE/RCe52QMhP0W9aenZiQTWWLTjGlwIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YLl36kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
O55Hz2O39OovPh0HQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zcqrouFEfKO
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTTlLiYDrFTOECQQDRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a6OJ6S24Z+zSr16oYHDTK5M6U2GhUlEXdyICt
b2OEqZwxAkEAyOlOjD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTkRIrZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8jO1TbH3qs11Zjcl9M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC2OLOBMzVSlRVA/5zgfNGO64snqteVdEavaxL3JEEVjmzw2yw2VNyMdumZlWzdv
SeQKxvUj3P3ms3GFpG8CQQCHom0+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cclqyYzNWBSxIeaBbpCOvjdqlDNcaX3aXupl
-----END RSA PRIVATE KEY-----
.
dx% list file
txcert
txkey
dx%
```

The DX appliance now has a certificate and key with which to perform SSL transactions.

Importing from ApacheSSL

The key and certificate locations are listed in the $APACHESSLROOT/conf/httpd.conf file. The default key is $APACHEROOT/certs/*.key. The default certificate is $APACHEROOT/certs/*.crt.Make note of these names and locations.

To import these files to the DX appliance, follow this example of copying and pasting the key and certificate files from the locations previously described.

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gT05MWTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQK
EwlERU1PIE9OTFkxEjAQBgNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE9OTFkwHhcNMDIwMzA1MjM1MzAxWhcN
MDIwMzA2MjM1MzAxWjCBizELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gT05M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQKEwlERU1PIE9OTFkxEjAQBgNV
BAsTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE9OTFkwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgL5Z5tcp8
HkubHFrpC1tub2CEANVBJsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAkoO0d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwHQYDVR0OBBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgbAwga2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQQGEwJYWDESMBAGA1UECBMJREVNTyBPTkxZ
MRIwEAYDVQQHEwlERU1PIE9OTFkxEjAQBgNVBAoTCURFTU8gT05MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRIwEAYDVQQDEwlERU1PIE9OTFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gT05MWYIBADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
WOFOiR7MsY64y5cbpMoGrfZ2qNgNKF+i6WLlmTfh4+1tKiCMnhTRPMcszjvwgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
.
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQWkhMCb+BrlT7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDJiQdQwOYlmaWQ7x5hSIE/RCe52QMhPOW9aenZiQTWWLTjGlwIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YLl36kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
O55Hz2O39OovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2wO9zcqrouFEfKO
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTTlLiYDrFTOECQQDRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a6OJ6S24Z+zSr16oYHDTK5M6U2GhUlEXdyICt
b2OEqZwxAkEAyOlOjD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTkRIrZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8jO1TbH3qs11Zjcl9M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC2OLOBMzVSlRVA/5zgfNGO64snqteVdEavaxL3JEEVjmzw2yw2VNyMdumZlWzdv
SeQKxvUj3P3ms3GFpG8CQQCHomO+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cclqyYzNWBSxIeaBbpCOvjdqlDNcaX3aXupl
-----END RSA PRIVATE KEY-----
.
dx% list file
txcert
txkey
dx%
```

The DX appliance now has a certificate and key with which to perform SSL transactions.

Importing from IIS 4 on Windows NT

The certificate file is in the directory that was specified when the certificate was downloaded.

1. Double-click the certificate file to open the viewer
2. Click the DETAILS tab
3. Click COPY to file. The Certificate Manager Export Wizard opens. Click NEXT.
4. Select the "Base 64 encoded X509" radio button. Click NEXT.
5. Specify a file name and location. Click NEXT.
6. Click FINISH.
7. Click OK when you see the successful completion notice.
8. Exit the Certificate Manager Export Wizard.
9. Close the certificate viewer.

The keys are located within the Key Ring (the key manager program). Follow these instructions to export a key:

1. Click the START button, point to Programs > Windows NT 4.0 Option Pack > Microsoft Internet Information Server, and click Internet Service Manager. The Microsoft Management Console will open.
2. Navigate to the Web site using the object list.
3. Right-click the Web site object and click PROPERTIES in the shortcut menu.
4. Click the DIRECTORY SECURITY tab.
5. Click EDIT in the Secure Communication panel.
6. Click KEY MANAGER.
7. Click the key to export.
8. In the Key menu, point to Export Key, and click BACKUP FILE.
9. Read the security warning and click OK.
10. Select a file location and enter a file name.
11. Click SAVE.
12. Exit the Internet Service Manager.

Exporting Key and Certificate Files to the DX Appliance:

Exporting the certificate

The IIS 4.0 certificate can be exported as “base64 encoded X509” format. Simply open the base-64 encoded file in an appropriate text editor and copy its contents to the clipboard. Then, at the DX appliance command prompt, type `capture file txcert`, and paste the certificate information that you copied. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txkey” (the name can be anything you choose).

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gTO5MWTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQK
EwlERU1PIE9OTFkxEjAQBgNVBAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE9OTFkwHhcNMDIwMzA1MjM1MzAxWhcN
MDIwMzA2MjM1MzAxWjCBizELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gTO5M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQKEwlERU1PIE9OTFkxEjAQBgNV
BAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE9OTFkwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgL5Z5tcp8
HkubHFrpC1tub2CEANVBJsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAkoO0d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwHQYDVR0OBBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgbAwga2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQQGEwJYWDESMBAGA1UECBMJREVNTyBPTkxZ
MRIwEAYDVQQHEwlERU1PIE9OTFkxEjAQBgNVBAoTCURFTU8gTO5MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRIwEAYDVQQDEwlERU1PIE9OTFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gTO5MWYIBADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
W0FOiR7MsY64y5cbpMoGrfZ2qNgNKF+i6WLlmTfh4+1tKiCMnhTRPMcszjvwgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
.
```

The certificate is now on the DX appliance.

Exporting the Key

First the IIS 4.0 key (iis4key.key) needs to be converted to the DX appliance format. Copy the file from the IIS machine to a UNIX machine in order to convert your key to base-64 encoded format. To do this, locate the key file and execute the following commands:

```
dx% hd iis4key.key | head
```

This will perform a hex dump and display the key file on-screen. Now find the byte pattern “30 82” in the key file, which should be located before the “private-key” text. Strip off everything before the “30 82” using the following commands:

```
dx% dd skip=1 bs=xx < iis4key.key > iis4key.key2
dx% openssl rsa -inform NET -in iis4key.key2 -out iis4key.b64
```

In the argument “`bs=xx`”, “`xx`” is the number of bytes you are stripping out. A byte is a two-digit pair of numbers. For example, “12 34 56 78” equals 4 (four) bytes, so you would enter “`bs=4`”. Typically the number of bytes will be around 30 (thirty). You now have a key in base-64 (`iis4key.b64`) encoding that can be used with the DX appliance.

Open the base-64 encoded file in a text editor and copy the contents. Then, at the DX appliance command prompt, type the command `capture file txkey`, press RETURN, and paste the contents of the file as follows. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txkey” (the name can be anything you choose).

```
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQWkhMCb+BrlT7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDJiQdQwOYlmaWQ7x5hSIE/RCe52QMhPOW9aenZiQTWWLTjGlwIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YLl36kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
O55Hz2O39OovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2wO9zcqrouFEfKO
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTTlLiYDrFTOECQQDRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a6OJ6S24Z+zSr16oYHDTK5M6U2GhUlEXdyICt
b2OEqZwxAkEAyOlOjD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTkRIrZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8jO1TbH3qs11Zjcl9M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC2OLOBMzVSlRVA/5zgfNGO64snqteVdEavaxL3JEEVjmzw2yw2VNyMdumZlWzdv
SeQKxvUj3P3ms3GFpG8CQQCHomO+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cclqyYzNWBSxIeaBbpCOvjdqlDNcaX3aXupl
-----END RSA PRIVATE KEY-----
.
```

Now verify that you have the certificate and key files:

```
dx% list file
  txcert
  txkey
  dx%
```

The DX appliance now has a certificate and key for SSL transactions.

Importing from IIS 5 on Windows 2000

Follow these steps to export a certificate and key from IIS 5 on Windows 2000.

1. Click the START button, point to Programs > Administrative Tools, and click Internet Service Manager. Alternately, open the Internet Service Manager in the Administrative Tools folder in the Control Panel.
2. Right-click the Web site object and click PROPERTIES in the shortcut menu.
3. Click the DIRECTORY SECURITY tab.
4. Click VIEW CERTIFICATE in the Secure Communications panel. The Certificate Viewer appears.
5. Click the DETAILS tab.
6. Click Copy to file. The Certificate Export Wizard appears. Click NEXT.
7. The Export Private Key panel appears.
8. Choose “YES, EXPORT THE PRIVATE KEY” option. Click NEXT.
9. The Export File Format panel appears.

10. Choose the PERSONAL INFORMATION EXCHANGE - PKCS#12 (PFX) option and any optional choices desired. Click Next.

11. The Password panel appears. Type in the password and confirm the password text boxes. Click NEXT.

12. The File to Export panel appears.

13. Type the path and file name in the File name text box or click Browse to select a location manually. Click NEXT.

14. Completing the Certificate Export Wizard panel appears.

15. Click FINISH.

Now the IIS 5.0 cert and key (iis5certkey.pfx) must be converted to base-64 encoded format. Move the files to a server that has OpenSSL installed and use the following command:

```
dx% openssl pkcs12 -nodes -in iis5certkey.pfx
```

This will print the file which contains the certificate and key on-screen. Scan the file for the relevant certificate and key information. The certificate information will look like:


```
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gT05MWTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQK
EwlERU1PIE9OTFkxEjAQBgNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE9OTFkwHhcNMDIwMzA1MjM1MzAxWhcN
MDIwMzA2MjM1MzAxWjCBizELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gT05M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQKEwlERU1PIE9OTFkxEjAQBgNV
BAsTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE9OTFkwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgL5Z5tcp8
HkubHFrpC1tub2CEANVBJsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAkoO0d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwHQYDVR0OBBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgbAwga2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQQGEwJYWDESMBAGA1UECBMJREVNTyBPTkxZ
MRIwEAYDVQQHEwlERU1PIE9OTFkxEjAQBgNVBAoTCURFTU8gT05MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRIwEAYDVQQDEwlERU1PIE9OTFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gT05MWYIBADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
W0FOiR7MsY64y5cbpMoGrfZ2qNgNKF+i6WLlmTfh4+1tKiCMnhTRPMcszjvwgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
```


And the key will look like:


```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQWkhMCb+BrlT7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDJiQdQwOYlmaWQ7x5hSIE/RCe52QMhPOW9aenZiQTWWLTjGlwIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YLl36kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
O55Hz2O39OovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zcqrouFEfKO
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTTlLiYDrFTOECQQDRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a6OJ6S24Z+zSr16oYHDTK5M6U2GhUlEXdyICt
b2OEqZwxAkEAyOlOjD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTkRIrZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8jO1TbH3qs11Zjcl9M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC2OLOBMzVSlRVA/5zgfNGO64snqteVdEavaxL3JEEVjmzw2yw2VNyMdumZlWzdv
SeQKxvUj3P3ms3GFpG8CQQCHom0+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cclqyYzNWBSxIeaBbpC0vjdqlDNcaX3aXupl
-----END RSA PRIVATE KEY-----
```

Leave the window open and note the location of this information. You will copy and paste it onto the DX appliance in the next step.

Exporting Key and Certificate Files to the DX Appliance

Open a SSH connection to the DX appliance. Copy and paste the key and certificate information you just noted into the DX appliance using the following steps

At the DX appliance command prompt, type the following command `capture file txcert`, then paste the certificate information. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file "txcert" (the name can be anything you choose).

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gTO5MWTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQK
EwlERU1PIE9OTFkxEjAQBgNVBAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE9OTFkwHhcNMDIwMzA1MjM1MzAxWhcN
MDIwMzA2MjM1MzAxWjCBizELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gTO5M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRIwEAYDVQQKEwlERU1PIE9OTFkxEjAQBgNV
BAsTCURFTU8gTO5MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE9OTFkwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgL5Z5tcp8
HkubHFrpC1tub2CEANVBJsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAkoO0d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwHQYDVR0OBBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgbAwga2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQQGEwJYWDESMBAGA1UECBMJREVNTyBPTkxZ
MRIwEAYDVQQHEwlERU1PIE9OTFkxEjAQBgNVBAoTCURFTU8gTO5MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRIwEAYDVQQDEwlERU1PIE9OTFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gTO5MWYIBADAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
W0FOiR7MsY64y5cbpMoGrfZ2qNgNKF+i6WLlmTfh4+1tKiCMnhTRPMcszjvwgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
.
```

Your certificate is now on the DX appliance.

Now, at the DX appliance command prompt, type the following command, `capture file txkey`, press ENTER, then paste the key information you noted previously. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file "`txkey`" (the name can be anything you choose).

```
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQWkhMCb+BrlT7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDJiQdQwOYlmaWQ7x5hSIE/RCe52QMhPOW9aenZiQTWWLTjGlwIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YLl36kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
O55Hz2O39OovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2wO9zcqrouFEfKO
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTTlLiYDrFTOECQQDRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a6OJ6S24Z+zSr16oYHDTK5M6U2GhUlEXdyICt
b2OEqZwxAkEAyOlOjD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTkRIrZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8jO1TbH3qs11Zjcl9M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC2OLOBMzVSlRVA/5zgfNGO64snqteVdEavaxL3JEEVjmzw2yw2VNyMdumZlWzdv
SeQKxvUj3P3ms3GFpG8CQQCHomO+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
```

```
fvJaD763cclqyYzNWBSxIeaBbpCOvjdqlDNcaX3aXupl
-----END RSA PRIVATE KEY-----
.
```

Now verify that you have the certificate and key files:

```
dx% list file
txcert
txkey
dx%
```

The DX appliance now has a certificate and key with which to perform SSL transactions.

Importing from iPlanet

The `pk12util` command available on the iPlanet server allows you to export certificates and keys from the internal database of iPlanet server and import them into the DX appliance. By default, `pk12util` uses certificate and key databases named `cert7.db and key3.db`.

To export a certificate and key from the iPlanet server, perform the following steps:

1. Go to the server_root/alias directory containing the databases.

   ```
   dx% cd server_root/alias
   ```

2. Add server_root/bin/https/admin/bin to your path.
3. Locate pk12util in server_root/bin/https/admin/bin.
4. Set the environment. For example:

```
On Unix: setenv LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}
On IBM-AIX: LIBPATH
On HP-UX: SHLIB_PATH
On NT, add it to the PATH
LD_LIBRARY_PATH server_root/bin/https/bin
```

You can find the path for your machine as server_root/https-admin/start.

1. Enter the `pk12util` command to view available options:

   ```
   dx% pk12util
   ```

2. Perform the actions required. For example, in Unix you would enter:

```
dx% pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P https-test-host]
```

3. Enter the database password.
4. Enter the `pkcs12` password.

To import the SSL key and certificate into the DX appliance, you must run the OpenSSL command on the file output from the `pk12util` utility as mentioned previously:

```
dx% openssl pkcs12 -in certpk12
```

This will print out the certificate and key in the base-64 encoded format, which you will then need to copy and paste onto the DX appliance using the `capture file` command. The above example assumes that `certpk12` was the output from the `pk12util` command.

Generating Keys and Certificates

GEN KEY

Usage: `gen key <key_file>`

The "`gen key file`" command is short for "generate private key." It generates a 1024-bit RSA private key. A filename MUST be specified, and the key is saved into that file.

Sample:

```
dx% gen key my.key
Saved as my.key...
dx%
```

GEN CSR

Usage: `gen csr <key_file> <csr_file>`

This command is short for "generate certificate signing request." It accepts a 1024-bit RSA private key, then prompts the user for information (i.e., country, organization name, common name, state, city, etc.), then creates a certificate signing request based upon the key and the user's input. The most important field is the "common name", which must match the DNS name of the cluster's listen address. The CSR should be sent to a Certificate Authority (like Verisign or Thawte) in exchange for an official certificate, which can then be imported into the DX appliance via the `capture file`. A filename MUST be specified and the CSR is saved to the file. The `list file` command can be used to view the CSR.

Sample:

```
dx% gen csr my.key my.csr
Please supply the requested information to form the Distinguished Name
(DN) incorporated in your certificate.

You may accept the default value shown in brackets by pressing enter,
or force a field to be blank by entering a single '.' and pressing enter.

Please note: to prevent security errors, the Common Name field should
match the host name (fully-qualified domain name) that browsers address
this machine as.

Country name (2 letter code) [US]:
State or province name (full name) [California]:
Locality name (eg, city) []:
Organization (company) name []:
Organizational unit name []:
Common name (advertised host name) [dx.juniper.net]:
Email address []:
```

```
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=US, ST=California, CN=dx.juniper.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:cf:c1:7e:d8:3c:68:be:26:9f:98:c0:07:d1:c9:
                    fb:57:80:d8:17:28:20:27:74:24:f3:5a:df:13:0a:
                    54:60:ba:39:5c:bf:8d:85:4e:56:14:b2:6c:26:03:
                    5d:92:80:f6:0b:44:4d:cc:d4:a4:99:11:6d:ce:a2:
                    bb:4c:b6:7d:24:75:ac:95:53:ae:2a:90:48:51:bf:
                    51:68:15:39:f5:4b:2c:7c:5e:50:6b:5b:f5:4a:5e:
                    d1:6f:60:a9:de:6e:96:ed:5c:95:e1:b0:33:97:b8:
                    d8:4c:78:7c:e6:9d:dd:68:76:50:97:c5:99:0c:43:
                    72:69:bc:9e:4e:ab:c7:a1:2b
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: md5WithRSAEncryption
        64:90:e2:c1:7a:41:c0:fd:51:4b:2d:79:71:43:69:9f:1d:82:
        80:54:67:45:5b:48:b1:71:c2:c3:51:e2:94:d7:a3:66:45:94:
        05:24:37:cb:33:09:4f:cb:4b:7c:66:6f:af:ac:a3:47:7c:19:
        71:42:7d:26:c8:bd:fc:6e:b2:2b:99:d0:24:53:d2:77:27:13:
        4f:ff:59:ff:f1:6a:c5:0e:d1:35:27:f0:4c:63:dc:50:22:e8:
        29:88:4b:a0:70:f0:1f:16:d5:bc:61:43:60:8a:e0:ff:f8:f6:
        df:f9:73:8c:81:46:77:67:50:30:df:6f:b4:62:76:36:8e:60:
        3a:00
-----BEGIN CERTIFICATE REQUEST-----
MIIBhDCB7gIBADBFMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEhMB
8GA1UEAxMYZXR4Mi5yZWRsaW5lbmV0d29ya3MuY29tMIGfMA0GCSqGSIb3DQEBAQUA
A4GNADCBiQKBgQDPwX7YPGi+Jp+YwAfRyftXgNgXKCAndCTzWt8TClRgujlcv42FTl
YUsmwmA12SgPYLRE3M1KSZEW3OortMtn0kdayVU64qkEhRv1FoFTn1Syx8XlBrW/VK
XtFvYKnebpbtXJXhsDOXuNhMeHzmnd1odlCXxZkMQ3JpvJ5Oq8ehKwIDAQABoAAwDQ
YJKoZIhvcNAQEEBQADgYEAZJDiwXpBwP1RSy15cUNpnx2CgFRnRVtIsXHCw1HilNej
ZkWUBSQ3yzMJT8tLfGZvr6yjR3wZcUJ9Jsi9/G6yK5nQJFPSdycTT/9Z//FqxQ7RNS
fwTGPcUCLoKYhLoHDwHxbVvGFDYIrg//j23/lzjIFGd2dQMN9vtGJ2No5gOgA=
-----END CERTIFICATE REQUEST-----
Saving as my.csr...
dx%
```

GEN SSC

Usage: `gen ssc <key_file> <ssc_file>`

This command is short for “generate self-signed certificate.” It accepts a 1024-bit RSA private key, then prompts the user for information (i.e., country, organization name, common name, state, city, etc.), then generates a self-signed certificate based upon the key and the user's input. The most important field is the “common name,” which must match the DNS name of the cluster's listen address. This certificate can be used on the DX appliance. The certificate will be “phony,” but it may be sufficient for a company's internal test lab. The filename MUST be specified, and the certificate is saved to that file.

Sample:

```
dx% gen ssc my.key my.ssc
Please supply the requested information to form the Distinguished
Name (DN) incorporated in your certificate.
```

```
You may accept the default value shown in brackets by pressing enter,
or force a field to be blank by entering a single '.' and pressing
enter.

Please note: to prevent security errors, the Common Name field should
match the host name (fully-qualified domain name) that browsers address
this machine as.

Country name (2 letter code) [US]:
State or province name (full name) [California]:
Locality name (eg, city) []:
Organization (company) name []:
Organizational unit name []:
Common name (advertised host name) [dx.juniper.net]:
Email address []:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=US, ST=California, CN=dx.juniper.net
        Validity
            Not Before: Apr  8 17:48:29 2002 GMT
            Not After : Apr  8 17:48:29 2003 GMT
        Subject: C=US, ST=California, CN=dx.juniper.net
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
                Modulus (1024 bit):
                    00:bc:64:f1:2c:a7:96:ee:d5:df:64:b2:f3:2a:a3:
                    99:e0:c3:41:ba:97:3c:dc:6c:15:ba:dc:b0:bc:a2:
                    5a:9c:78:12:1c:c7:22:ff:16:f2:d5:8a:8f:0b:2a:
                    f8:02:f6:35:b1:5b:b5:9d:f9:35:4c:36:0d:6b:bc:
                    0e:ce:0a:cd:6a:b9:bd:2e:db:e6:82:b6:c2:c8:6a:
                    4c:5f:d7:e7:78:cc:8d:2e:22:c9:15:52:df:97:ae:
                    71:e8:c8:c1:b3:4c:13:6d:d7:01:f7:1b:4a:4e:51:
                    bf:dc:78:c1:1c:96:b2:da:33:fa:88:20:a0:5e:ec:
                    9f:57:73:59:5a:90:4b:23:3b
                Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
        89:d9:f0:a4:d0:ec:5c:cb:6a:24:28:19:8d:95:1d:ff:27:bc:
        f2:0f:f7:97:d4:35:a1:75:6e:8a:28:ce:17:55:01:ed:36:95:
        c6:28:01:24:11:46:23:ee:da:1d:f5:53:82:65:18:84:dd:99:
        33:c5:9b:62:fa:af:d9:29:28:32:13:a0:47:3d:74:82:ec:d1:
        04:98:cb:29:11:ac:6e:21:39:37:3f:a7:70:86:0b:30:43:32:
        24:62:1d:40:d1:0c:d0:c5:cc:74:24:d7:47:2b:e9:7f:f6:fd:
        5f:68:08:88:30:40:44:5e:07:5f:f3:e5:fc:ed:fd:c9:d3:e3:
        a1:6b
-----BEGIN CERTIFICATE-----
MIIB/jCCAWegAwIBAgIBADANBgkqhkiG9w0BAQQFADBFMQswCQYDVQQGEwJVUzETMB
EGA1UECBMKQ2FsaWZvcm5pYTEhMB8GA1UEAxMYZXR4Mi5yZWRsaW5lbmV0d29ya3Mu
Y29tMB4XDTAyMDQwODE3NDgyOVoXDTAzMDQwODE3NDgyOVowRTELMAkGA1UEBhMCVV
MxEzARBgNVBAgTCkNhbGlmb3JuaWExITAfBgNVBAMTGGV0eDIucmVkbGluZW5ldHdv
cmtzLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAvGTxLKeW7tXfZLLzKq
OZ4MNBupc83GwVutywvKJanHgSHMci/xby1YqPCyr4AvY1sVu1nfk1TDYNa7wOzgrN
arm9LtvmgrbCyGpMX9fneMyNLiLJFVLfl65x6MjBs0wTbdcB9xtKTlG/3HjBHJay2j
P6iCCgXuyfV3NZWpBLIzsCAwEAATANBgkqhkiG9w0BAQQFAAOBgQCJ2fCk0Oxcy2ok
KBmNlR3/J7zyD/eX1DWhdW6KKM4XVQHtNpXGKAEkEUYj7tod9VOCZRiE3ZkzxZti+q
/ZKSgyE6BHPXSC7NEEmMspEaxuITk3P6dwhgswQzIkYh1AOQzQxcx0JNdHK+l/9v1f
aAiIMEBEXgdf8+X87f3J0+Ohaw==
-----END CERTIFICATE-----
Saving as my.ssc...
dx%
```

SSL Ciphersuite Details

The following SSL ciphersuites are available on the DX appliance.

Table 11: SSL Ciphersuites

Ciphersuite	Description
Common SSL Ciphers RC4-MD5 RC4-SHA EXP-RC4-MD5 EXP-RC2-CBC-MD5 EXP1024-RC4-MD5 EXP1024-RC2-CBC-MD5	The fastest ciphersuites from both the Strong and Export groups.
Strong SSL Ciphers RC4-MD5 RC4-SHA DES-CBC3-MD5 DES-CBC3-SHA AES256-SHA AES128-SHA IDEA-CBC-SHA IDEA-CBC-MD5	The highest-security ciphersuites that are suitable for use in USA.
Export SSL Ciphers EXP-RC4-MD5 EXP-RC2-CBC-MD5 EXP1024-RC4-MD5 EXP1024-RC2-CBC-MD5 DES-CBC-MD5 DES-CBC-SHA	Lower-security ciphersuites that are suitable for export
All SSL Ciphers RC4-MD5 RC4-SHA DES-CBC-MD5 DES-CBC-SHA DES-CBC3-MD5 DES-CBC3-SHA AES256-SHA AES128-SHA IDEA-CBC-SHA IDEA-CBC-MD5 EXP-RC4-MD5 EXP-RC2-CBC-MD5 EXP1024-RC4-MD5 EXP1024-RC2-CBC-MD5	Strong and Export

Forcing Clients to use HTTPS with Cluster Redirection (Auto SSL)

The Cluster Redirection feature allows you to redirect requests from a browser to a new location or redirect requests using a different protocol (HTTP or HTTPS). Some examples of uses for this feature are:

- Redirecting all requests coming in via HTTP on port 80 to the *same page* using HTTPS on port 443
- Redirecting all requests coming in via HTTP on port 80 to a *new page* using HTTP on port 80
- Redirecting all requests coming in via HTTP on port 80 to a *new page* using HTTPS on port 443

To redirect client requests, the DX appliance responds with the HTTP 302 “temporarily moved” response code in compliance with RFC 2616. The response also contains the new location in an HTTP Location header which both HTTP 1.0 and HTTP 1.1-compliant clients recognize.

NOTE: When using Auto SSL to redirect requests from HTTP:// to HTTPS://, any hard-coded HTTP links in the content page will get redirected to HTTP with an HTTP 302 redirect message. The 302 redirect message will be in clear using HTTP, so the browsers may see warnings which indicate that the page contains “secure and non-secure items.” The only non-secure items are the HTTP 302 redirect messages.

EXAMPLE: Configuring Cluster Redirection to Redirect HTTP Requests to HTTPS

The following example shows how to use a Redirector to redirect all incoming HTTP requests on port 80 to be HTTPS requests on port 443 of the same VIP. The VIP in this example is 205.178.13.100.

1. Add a redirector with the same VIP as the cluster where you wish to redirect requests:

```
dx% add redirector
dx% set redirector 1 listen vip 205.178.13.100
```

2. Set the listen port for the redirector to 80:

```
dx% set redirector 1 listen port 80
```

3. Set the target port for the redirector to 443:

```
dx% set redirector 1 port 443
```

4. Set the redirector protocol to HTTPS. This will instruct the browser to use HTTPS when connecting to the redirected location:

```
dx% set redirector 1 protocol https
```

5. Set the host where the redirector will direct requests. You can enter either the fully-qualified domain name or the IP address of the host where requests should be redirected. You should NOT use an IP address if either of the following is true:

- The VIP is a private IP address.
- Multiple DX appliances are being load balanced by an upstream load balancer

```
dx% set redirector 1 host www.mysite.com
dx% set redirector 1 host 205.178.13.100
Enable the Redirector:
dx% set redirector 1 enabled
```

6. OPTIONAL: Configure the redirector to redirect requests to a custom URL. By default, clients are redirected to the same page initially requested at the new location.

If you would like to send the browser to a different page, such as a secure login page, you must set a custom URL and set the URL method to custom. The custom URL must be configured before the URL method.

```
dx% set redirector 1 customURL "/secure_login.html"
dx% set redirector 1 URLmethod custom
```

Configuring SSL Client Authentication

Overview

SSL client authentication lets the DX appliance accept connections only from clients possessing the proper credentials; in this case, an SSL Client Certificate designated as valid. SSL client authentication can be enabled on a VIP/cluster basis. Clients will be unable to access restricted information unless they possess a valid SSL Client Certificate. Unauthorized clients do not have the opportunity to make an HTTP request before their connection is terminated.

To authenticate clients, the DX appliance can use root certificates and corresponding certificate revocation lists (CRL) issued by well-known, trusted Certificate Authorities (CA), such as Verisign, Thawte, etc. or certificates and CRLs from an in-house CA.

In an Enterprise environment, it is likely that there will be an in-house CA that provides certificates for client authentication. The in-house CA can publish client certificates and intermediate CA certificates that allow other organizations within the Enterprise environment to act as their own CAs.

It is important to note that the DX appliance DOES NOT perform the following tasks:

- The DX appliance does not act as a CA.
- The DX appliance does not generate its own CA certificate or CRLs for that certificate.
- The DX appliance does not generate client certificates.

These items must be generated outside of the DX appliance and imported to the DX appliance for use in the client authentication process. More detailed information on how the DX appliance stores and presents CA certificates and CRLs is outlined in the following sections.

Certificate Authority (CA) Certificate Presentation

The DX appliance can store and present (i.e., advertise) one or more valid CA certificates to the client during the SSL handshake. The advertised CA certificate(s) can either be root certificates, from a well-known trusted CA, in-house CA certificates, or intermediate CA certificates. The DX appliance is capable of storing multiple CA certificates per VIP. This allows you to present one or more CA certificate to clients based upon the VIP for the client connection.

All certificates listed in the advertised CA certificate file must be in base64-encoded format. The following is an example of this format:


```
-----BEGIN CERTIFICATE-----
MIICpDCCAg0CAQEwDQYJKoZIhvcNAQEEBQAwgbwxCzAJBgNVBAYTAlVTMRMwEQYD
VQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhDYW1wYmVsbDEZMBcGA1UEChMQUmVk
bGluZSBOZXR3b3JrczEUMBIGA1UECxMLRW5naW5lZXJpbmcxKjAoBgNVBAMTIUVu
Z2luZWVyaW5nIENlcnRpZmljYXRlIEF1dGhvcml0eTEoMCYGCSqGSIb3DQEJARYZ
ZW5nY2FAcmVkbGluZW5ldHdvcmtzLmNvbRcNMDIxMDMxMDExMjI4WhcNMDIxMTA3
MDExMjI4WjAUMBICAQMXDTAyMTAzMTAxMDAxM1qgggEEMIIBADCB/QYDVR0jBIH1
```

```
MIHygBSU0vjIlDn+HXdpi22BMTpgBfFLrKGB1qSB0zCB0DELMAkGA1UEBhMCVVMx
EzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCENhbXBiZWxsMRkwFwYDVQQK
ExBSZWRsaW5lIE5ldHdvcmtzMSMwIQYDVQQLExpSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eTEvMC0GA1UEAxMmUmVkbGluZSB0ZXR3b3JrcyBDZXJ0aWZpY2F0ZSBB
dXRob3JpdHkxKDAmBgkqhkiG9w0BCQEWGXJsbmNhQHJlZGxpbmVuZXR3b3Jrcy5j
b22CAQEwDQYJKoZIhvcNAQEEBQADgYEAhudjWq+t1tx0cJa63H36eQgBRew6QNnK
PtDdC5Lojhu9dETmR+GKzalYyyD0Kmzl/QIx4GFWthNRXoUYWXwW/KWgayulGzru
JFbdQA0O4YiXYL9EeAWHXwhnOH+RHGtE+qjJFlYhXX3lonnQKyvKsuKxfbG7Nmku
jrnc42BgWuQ=
-----END CERTIFICATE-----
```


Trusted Certificate Authority (CA) Certificate Storage

The DX appliance can also maintain a list of CA certificates that are considered “trusted.” These trusted certificates are used to validate the certificate chain presented by the user. While the advertised list of certificates may only comprise a portion of the user's certificate chain, the trusted list must comprise the entire certificate chain for successful client authentication.

A certificate chain is a list of certificates formed by referring to each issuer of a certificate. For example, if root CA “Trusted Certs, Inc.” issues an intermediate CA certificate to “Company X”. Company X, in turn, issues a client certificate to employee Alice, then a certificate chain is formed from the root CA to the intermediate CA to the employee.

If Alice presents a certificate to the DX appliance that advertises Company X's intermediate CA certificate, then Alice can supply her client certificate for authentication. As part of the authentication process, the DX appliance will walk the certificate chain all the way back to the root CA certificate validating each one along the way.

In order to accomplish this, the trusted CA certificate file must contain not only Company X's intermediate CA certificate, but also the root CA certificate of Trusted Certs, Inc. If a trusted CA certificate file is not specified by the user, then a default trusted list of certificates is used by the DX appliance. This list is composed of all the major well-known CAs. Note that the list of trusted CA certificates does not include the client's certificates.

All certificates listed in the trusted CA certificate file must be in base64-encoded format.

Certificate Revocation List (CRL)

The DX appliance will terminate an SSL handshake if a client's certificate is present in a customer-specified Certificate Revocation List (CRL). One CRL may exist per entry in the trusted CA certificate file. A CRL is not required for activating SSL Client Authentication on a particular VIP.

The CRL must be in base64-encoded format. An example of this format is as follows (note that the header and trailer must match exactly as shown):


```
-----BEGIN X509 CRL-----
MIICpDCCAg0CAQEwDQYJKoZIhvcNAQEEBQAwgbwxCzAJBgNVBAYTAlVTMRMwEQYD
VQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhDYW1wYmVsbDEZMBcGA1UEChMQUmVk
bGluZSBOZXR3b3JrczEUMBIGA1UECxMLRW5naW5lZXJpbmcxKjAoBgNVBAMTIUVu
Z2luZWVyaW5nIENlcnRpZmljYXRlIEF1dGhvcml0eTEoMCYGCSqGSIb3DQEJARYZ
ZW5nY2FAcmVkbGluZW5ldHdvcmtzLmNvbRcNMDIxMDMxMDExMjI4WhcNMDIxMTA3
```

```
MDExMjI4WjAUMBICAQMXDTAyMTAzMTAxMDAxM1qgggEEMIIBADCB/QYDVR0jBIH1
MIHygBSU0vjIlDn+HXdpi22BMTpgBfFLrKGB1qSB0zCB0DELMAkGA1UEBhMCVVMx
EzARBgNVBAgTCkNhbGlmb3JuaWExETAPBgNVBAcTCENhbXBiZWxsMRkwFwYDVQQK
ExBSZWRsaW5lIE5ldHdvcmtzMSMwIQYDVQQLExpSb290IENlcnRpZmljYXRlIEF1
dGhvcml0eTEvMC0GA1UEAxMmUmVkbGluZSBOZXR3b3JrcyBDZXJ0aWZpY2F0ZSBB
dXRob3JpdHkxKDAmBgkqhkiG9w0BCQEWGXJsbmNhQHJlZGxpbmVuZXR3b3Jrcy5j
b22CAQEwDQYJKoZIhvcNAQEEBQADgYEAhudjWq+t1tx0cJa63H36eQgBRew6QNnK
PtDdC5Lojhu9dETmR+GKzalYyyD0Kmzl/QIx4GFWthNRXoUYWXwW/HWgayulGzru
NFbdQA0O4YiXYL9EeAWHXwhnOH+RHGtE+qjJFlYhXX3lonnQKyvKsuKxfbG7Nmku
jrnc42BgWuQ=
-----END X509 CRL-----
```

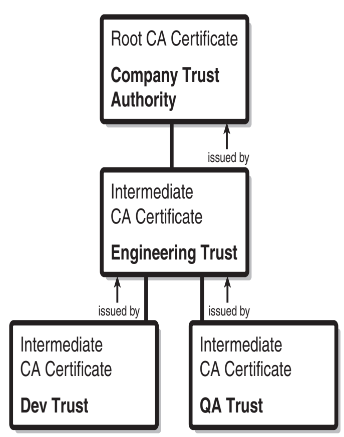

If a CRL is present, the client must satisfy the following three criteria in order to connect to the DX appliance and make requests.

- The client must possess a private key corresponding to the client certificate.
- The client certificate's certificate chain must be valid.
- The client's certificate must not exist within any Certificate Revocation List (CRL) corresponding to any certificate in the certificate chain.

Example of Chain Certificates and CRLs

Assume that you have a root CA known as “Company Trust Authority.” It is a root CA because its certificate is self-signed. That is, the issuer and the subject of the certificate are the same. Then let's assume that you create an intermediate CA called “Engineering Trust” whose certificate has been issued by the Company Trust Authority. Finally, let's assume that you have two additional intermediate CAs, “Development Trust” and “QA Trust” whose certificates have been signed by Engineering Trust. This effectively creates the certificate chain shown in Figure 41.

Figure 41: SSL Certificate Chain

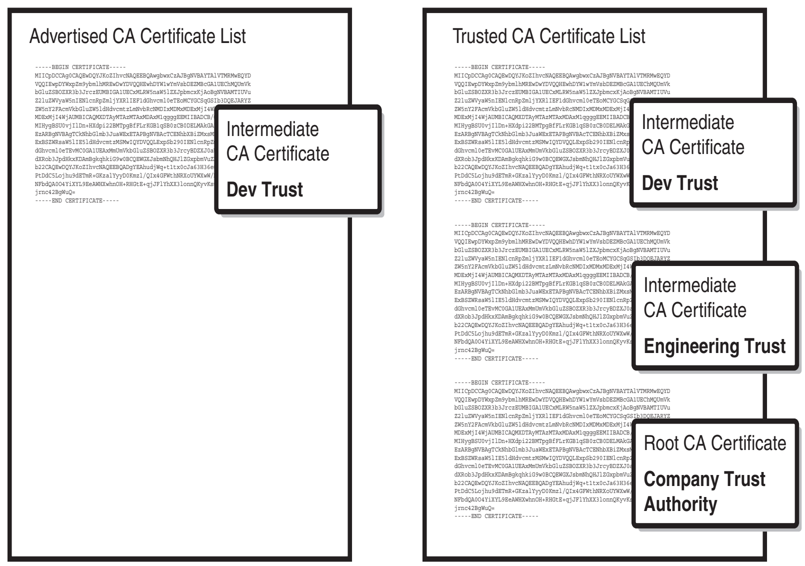


Based upon this certificate chain, an organization can issue certificates to various clients. In this example, you can issue a certificate to employee Alice from the Development Trust CA, and a certificate to Bob from the QA Trust CA. Note that in this case, both client certificates have been signed by intermediate CAs.

If you want to configure a DX appliance such that only those who have certificates from Development Trust are allowed access to the content available through that

DX appliance, then you would set up an advertised CA certificate list with the Development Trust CA certificate. However, our trusted CA certificate file would have entries for the Development Trust CA, the Engineering Trust CA, and the Company Trust Authority CA. Our arrangement would be something depicted in the following diagram. All entries are in base64-encoded format. Refer to Figure 42.

Figure 42: SSL Advertised and Trusted Lists

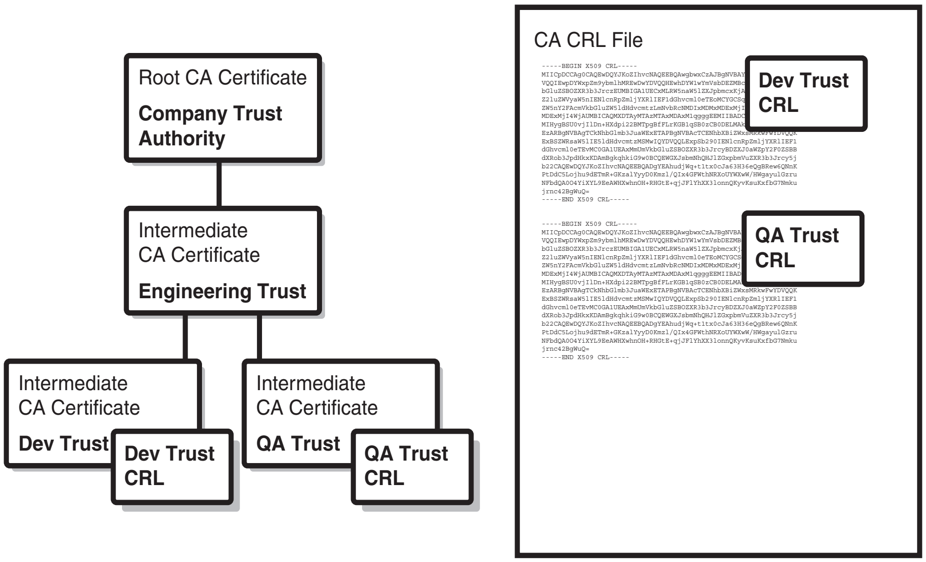


This arrangement is required because in order to satisfy the successful client authentication SSL handshake, the DX appliance must be able to validate the certificate chain all the way back to a root CA that it considered trusted. Note that each entry in the certificate files will be in base64-encoded format.

Client certificates may need to be invalidated from time to time. For example, if an employee who is issued a client certificate leaves the company, you need to establish a mechanism whereby the invalidated certificate is stored in some way relative to the “trusted list” of CA certificates. This is precisely the purpose of the CRL.

When a CRL exists for a certificate in the certificate chain, that CRL is consulted to make sure that the client's certificate is not on that list. Note that entries in a CRL are only certificates that have been issued by that CA. For the previous example, you might have this arrangement with regard to CRLs. Note that the CRL entries are also in base64-encoded format. Refer to Figure 43.

Figure 43: SSL In-House Control



DXSHELL Commands for SSL Client Authentication

Use these commands for SSL Client Authentication.

To enable SSL Client Authentication for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth enabled
```

To disable SSL Client Authentication for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth disabled
```

To configure the Advertised CA certificate file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth cacertfile <filename>
```

To configure the CRL file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth cacrlfile <filename>
```

To configure the trusted CA certificate file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth catrustfile <filename>
```

To clear the Advertised CA certificate file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth cacertfile
```

To clear the CA CRL file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth cacrlfile
```

To clear the trusted CA certificate file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth catrustfile
```

To display configurations for the client authentication, type the command:

```
dx% show cluster <name> listen ssl clientauth
```

To display the configuration value for the advertise CA certificate file, type the command:

```
dx% show cluster <name> listen ssl clientauth cacertfile
```

To display the configuration value for the CA CRL file, type the command:

```
dx% show cluster <name> listen ssl clientauth cacrlfile
```

To display the configuration value for the trusted CA certificate file, type the command:

```
dx% show cluster <name> listen ssl clientauth catrustfile
```

Browsers that Poorly Support SSL Client Authentication

Certain browsers do not have stable SSL client authentication implementations and thus their interoperability with this feature is unpredictable and not recommended. The browsers that exhibit this behavior are:

- Netscape 4.x
- Opera

Specifying Your Own List of SSL Ciphersuites

You can specify a file containing a list of SSL ciphersuites to configure an SSL cluster or redirector.

Capturing a Cipherfile

The cipherfile can be captured using the `capture file` command. It should contain a list of ciphersuites that conform to the OpenSSL standard. A sample list looks like:

```
RC4-MD5:MEDIUM:!DH:HIGH:!EXPORT56:-AES256-SHA
```

These commands support this feature:

```
% set cluster 1 listen ssl cipherfile <filename>
% set cluster 1 listen ssl ciphersuite file
% show cluster 1 listen ssl cipherfile
% show cluster 1 listen ssl cipherlist

% set cluster 1 target ssl cipherfile <filename>
% set cluster 1 target ssl ciphersuite file
% show cluster 1 target ssl cipherfile
% show cluster 1 target ssl cipherlist

% set redirector 1 listen ssl cipherfile <filename>
% set redirector 1 listen ssl ciphersuite file
% show redirector 1 listen ssl cipherfile
% show redirector 1 listen ssl cipherlist
```

If the ciphersuite is no file, then the cipherfile is ignored.

If SSL is enabled and a write is done, then the DXSHELL will validate the cipherfile in the same way that OpenSSL validates a ciphersuite list. OpenSSL is very lenient, but if OpenSSL does not complain, then DXSHELL will not either. For example, if cipherfile is set to demokey, OpenSSL will allow it because the first line "`-----BEGIN RSA PRIVATE KEY-----`" has a valid "RSA" keyword in it.

The "`show. cipherlist`" commands are provided so the user can confirm the actual list of ciphersuites to be used. Showing the cipherlist will print out a detailed line for each ciphersuite, showing the name, version, key exchange, authentication, encryption, and hash method.

NOTE: The "`show. cipherlist`" commands have no tab-completion because there is no way to distinguish a cipherfile from any other file.

NOTE: There is no WebUI support for specifying a cipherfile.

Some sample commands to configure a cipherfile are:

```
% capture file myciphers
Enter file. End with . on a blank line.
RC4-MD5:MEDIUM:HIGH:!EXPORT56
.

% set cluster 1 listen ssl ciphersuite file
```

```
(*)% set cluster 1 listen ssl cipherfile myciphers
(*)% write
% show cluster 1 listen ssl cipherlist
Cipherlist:
RC4-MD5                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
RC4-MD5                 SSLv2 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
AES128-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
IDEA-CBC-SHA            SSLv3 Kx=RSA      Au=RSA  Enc=IDEA(128) Mac=SHA1
RC4-SHA                 SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=SHA1
IDEA-CBC-MD5            SSLv2 Kx=RSA      Au=RSA  Enc=IDEA(128) Mac=MD5
RC2-CBC-MD5             SSLv2 Kx=RSA      Au=RSA  Enc=RC2(128)  Mac=MD5
AES256-SHA              SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
DES-CBC3-SHA            SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
DES-CBC3-MD5            SSLv2 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=MD5
%
```

The SSL AppRules Feature

A variable has been added to the Application Rules to support this feature:

- `ssl_cipher_bits`

along with two new test operators:

- `less_than`
- `greater_than`

The new test operators will only work with the `ssl_cipher_bits` test variable and the `ssl_cipher_bits` test variable will only work with Request Sentry rules.

The general usage to form a complete test condition is as follows:

```
RS: ssl_cipher_bits <less_than|greater_than> “<bit_length>”
```

where the `<bit_length>` can be an integer value between 0 and 1024. Typical values in the real world would be 40, 56, or 128.

This test condition can be used as a means of redirecting clients that do not have sufficiently strong browsers to a web page that would instruct them on how to download such a page. For example:

```
RS: ssl_cipher_bits less_than “128” then redirect
“http://browserupgrade.mysite.com/mysite/upgrade.html”
```

For additional information on Application Rules, see “Application Rules Syntax” on page 245.

Chapter 12
Logging the Client's IP

This chapter describes logging the client's IP into the DX Application Acceleration Platform, discussing the following topics:

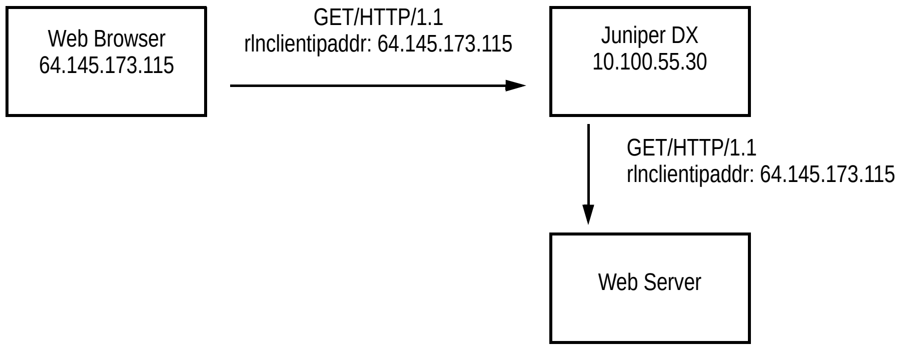
Overview

The DX appliance acts as a proxy to all target hosts (web servers, caches, etc.) so the IP sent to target hosts is the DX appliance's IP (refer to Figure 44). If logging is client cookie-based, no changes are required. All client cookie information will be sent from the DX appliance to the target hosts with each request. For logging configurations that record the origin client IP, the DX appliance offers two options.

- For site administrators who need a common log or combined log format, the DX appliance can compile the log information and send it to a Master logging machine running SYSLOG with the appropriate security enhancements.
- Alternatively, the DX appliance can be configured to record the client IP address for each web request in a custom HTTP header before forwarding the request to the web servers. For details on configuring your server to recognize this custom HTTP header, refer to the specific section in this chapter that matches your host type or server platform.

To ensure that the client IP address is not obscured from the backend servers when using this logging method, the DX appliance records the client IP address for each web request in a custom HTTP header before forwarding the request to the web servers.

Figure 44: The Flow of IP Address Information Between the Client, DX Appliance, and Server



Compiling Log Information on a Master Logging Machine

To have the DX appliance compile log information and send it to a Master logging machine.

- From the DXSHELL.

1. Provide the DX appliance with a log host:

   dx% **set cluster <name> log host <host IP>**

2. Turn logging on with the command:

   dx% **set cluster <name> log enabled**

- From the WebUI:

1. Open the DX appliance Settings page in the WebUI, and click on the "LOGGING" option.
2. Locate the Log Host Settings near the bottom of the page.
3. Enter the IP address of the Log Host and select the "ENABLED" option next to "LOGGING."
4. Click the SAVE button at the bottom of the page to save and apply changes.

Logging Client IP on the Webserver with a Custom Header

To pass client IP information on to web servers in the HTTP header.

1. You must use the DXSHELL command line to set the name of the header attribute that will contain the origin client's IP address. In the DXSHELL, enter the command:

   dx% **set server customiplogheader <header>**

where the `<header>` is a unique string similar to standard header attributes such as "Accept" or "User-Agent." For example, entering rlnclientipaddr as a

customiplogheader value would add the line below to the HTTP headers the DX appliance sends the web servers it is accelerating:

rlnclientipaddr: 64.145.173.115

2. Configure the logging utility running on your web servers to look for the custom HTTP header attribute and to record its value along with the other logging data.

Configuring Logging with Apache

To configure logging with Apache from the DXSHELL command-line interface on the DX appliance, use the command:

```
dx% set server customiplogheader rlnclientipaddr
```

to set the custom header field in which the DX appliance will insert the origin client's IP address.

On the Apache server, make sure that `mod_log_config.so` is enabled. It is typically enabled by default on Apache 1.3.x, but it is best to check.

Now you need to edit `http.conf,` making sure that:

```
CustomLog /var/log/httpd/access_log combined
```

is set (it is set by default). Then change:

```
LogFormat “%h %l %u %t \"%r\" %>s %b \"%{Referer}i\” \"%{User-Agent}i\""combined
```

to

```
LogFormat “%{rlnclientipaddr}i %l %u %t \"%r\" %>s %b \”%{Referer}i\"\"%{User-Agent}i\”” combined
```

where %{rlnclientipaddr}i matches the string that you set on the DX appliance box. Restart Apache and your logs will now reflect these changes.

Now instead of the DX appliance IP address (10.100.1.66) in the example below appearing in the server's access log:

```
10.100.1.66 - - [13/Aug/2001:12:11:25 -0700] “GET / HTTP/1.1” 304 - “-”
“Mozilla/4.77 [en] (Win95; U)”
```

the origin client's IP address, 192.168.3.87, will be recorded.

```
192.168.3.87 - - [13/Aug/2001:12:19:08 -0700] “GET / HTTP/1.1” 304 - “-”
“Mozilla/4.77 [en] (Win95; U)”
```

Configuring Logging with IIS

The file `rllog.dll` is an Internet Server API (ISAPI) filter that can be installed on an IIS server in order to log the real client IP address instead of the DX appliance IP address. Juniper Networks distributes two versions of `rllog.dll`:

- `rllog.dll` is compiled with the “Default” execution priority.
- `rllog.dll_HIGH_PRIORITY` is compiled with a HIGH execution priority.

Both are included in the bundle available at the Juniper Networks Technical Support site.

In most cases, you should use the “Default” priority version of `rllog.dll`. The “High” priority version forces IIS to execute the `rllog.dll` filter before other ISAPI filters. This is useful when other filters need access to the real client IP address that the `rllog.dll` inserts into the log structure PHTTP_FILTER_LOG in place of the DX appliance IP address.

1. The DX appliance must be configured to send the client IP address to IIS in a special HTTP header. From the DXSHELL command-line interface on the DX appliance, use the command:

   ```
   dx% set server customiplogheader rlnclientipaddr
   ```

to set the custom header field in which the DX appliance will insert the origin client's IP address.

2. Select which version of the `rllog.dll` version you are using and copy it to %SYSTEMROOT%\system\ on the IIS server.

3. %SYSTEMROOT% is the directory that contains Windows system files, commonly C:\WINNT, so in most cases you should copy `rllog.dll` to:

   C:\WINNT\system\

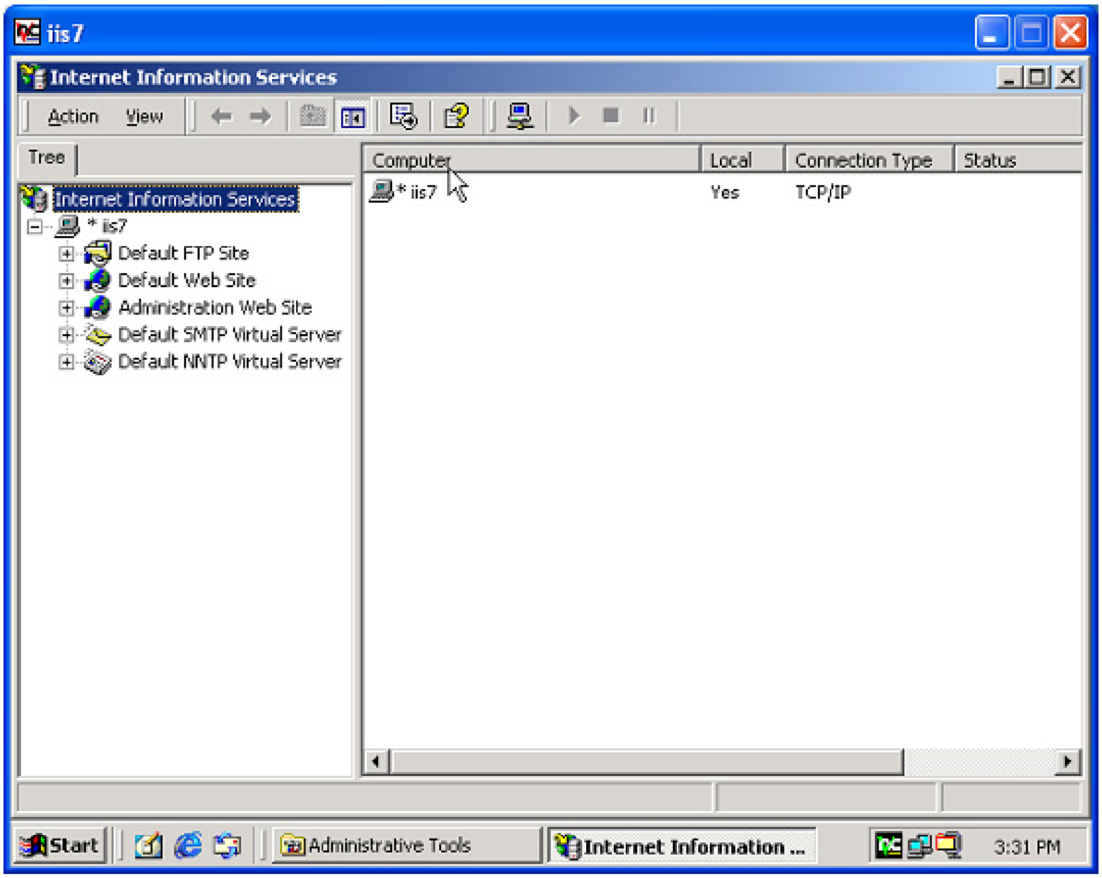
NOTE: The DEFAULT priority version of the filter, rllog.dll, is suitable for most uses.

4. If you have decided to use `rllog.dll_HIGH_PRIORITY`, rename it to `rllog.dll`.

5. Configure the IIS server to use the `rllog.dll` ISAPI filter.

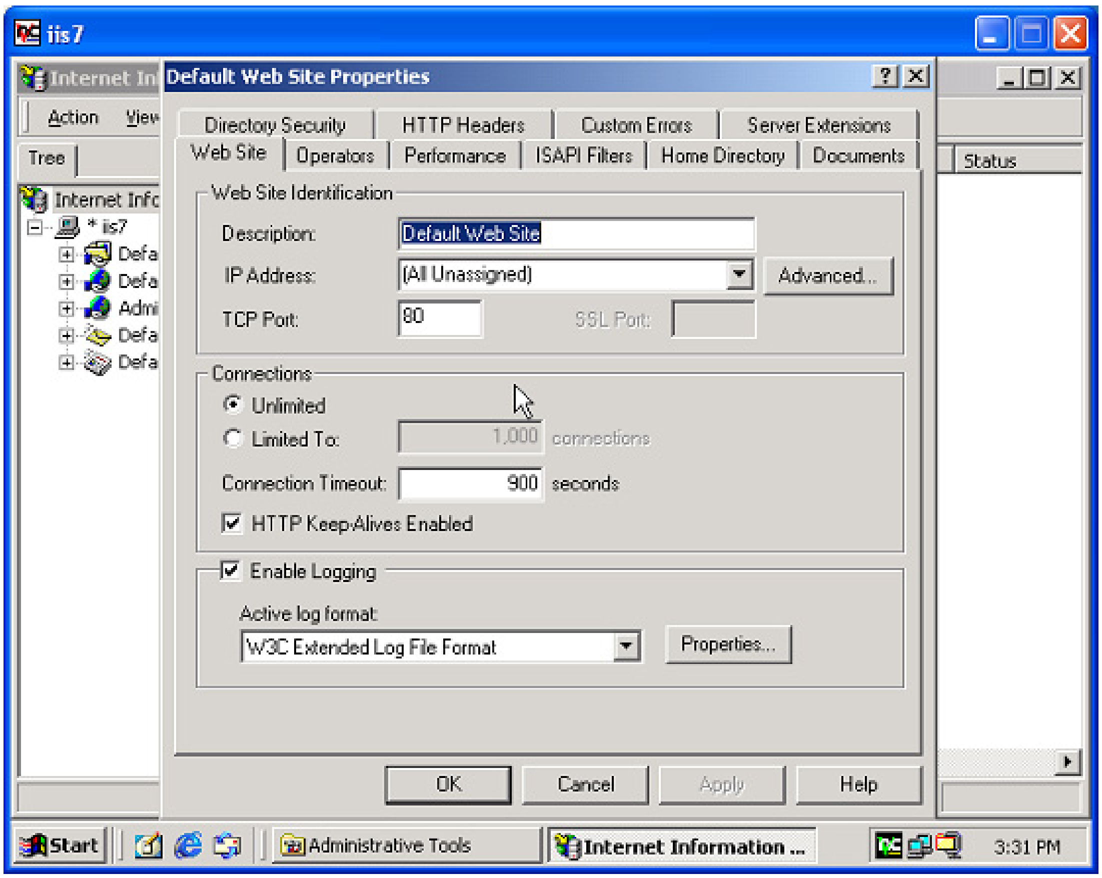
   - Open the Internet Information Services Administrator window (refer to Figure 45).

Figure 45: The IIS Administrator Window



- In the left hand column of the IIS Administrator window, find the name of the website for which you wish to install the filter. Right click on the website's name to bring up a contextual menu and select PROPERTIES from the contextual menu (refer to Figure 46).

Figure 46: The Web Site's Properties Dialog Box



- In the PROPERTIES dialog, select the “ISAPI Filters” tab and click the ADD button to add a new filter.

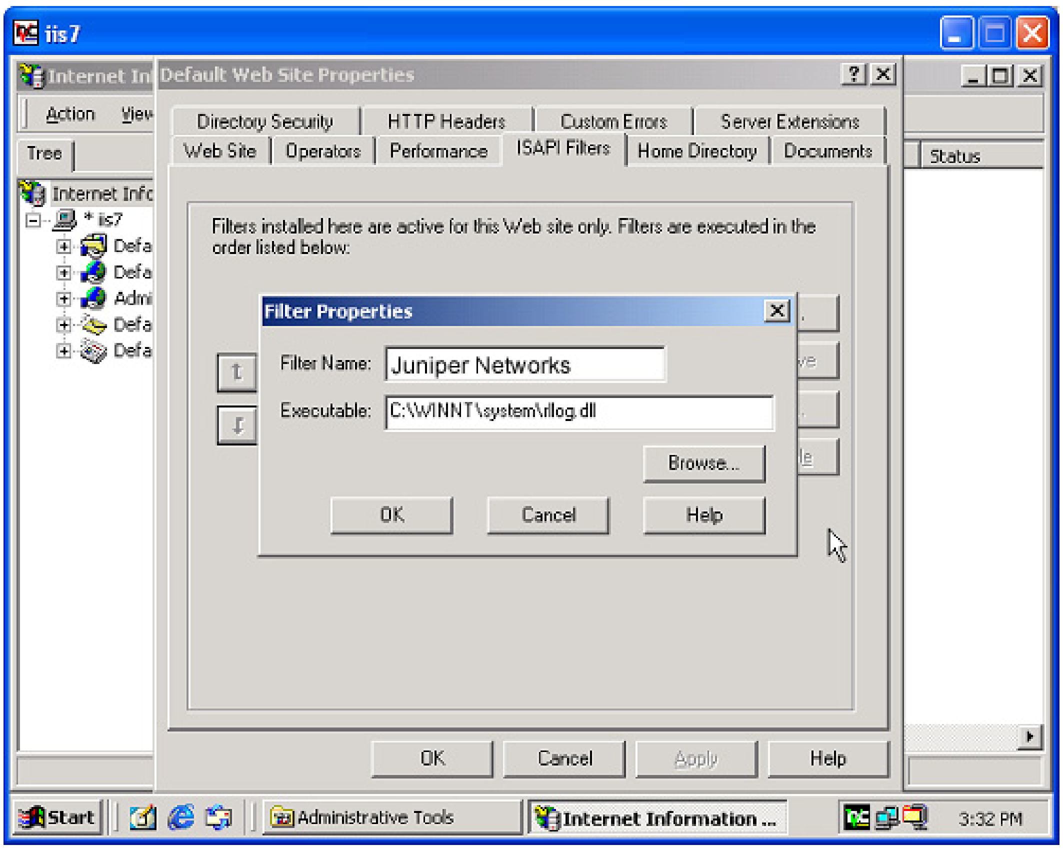
  - For “Filter Name” enter:

  `Juniper Networks`

  - For “Executable” enter:

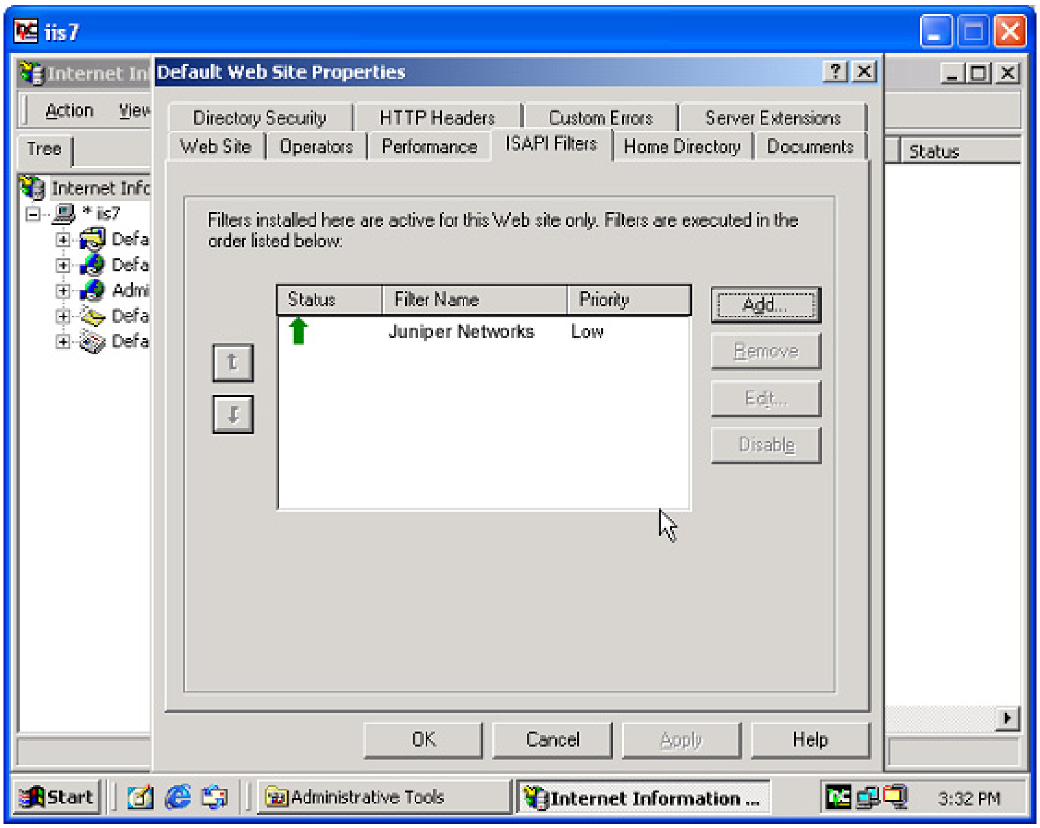
  `C:\WINNT\system\rllog.dll`
  or press the “BROWSE. ” button to locate `rlllog.dll` on the computer (refer to Figure 47).

Figure 47: Adding the `rllog.dll` Filter



- Click the OK button to save your changes and close the PROPERTIES dialog box.

Figure 48: After Adding the Juniper Networks `rllog.dll` Filter



- Use the stop and start buttons at the top of the IIS Administrator window to stop and then restart the web service.

Configuring Logging with Resin

In the file `resin.conf`, replace the line:

```
<access-log id='log/access.log' format='%h %l %u %t “%r” %s %b “%{Referer}i” “%{User-Agent}i”'/>'
```

with the line:

```
<access-log id='log/access.log' format='%{rlnclientipaddr}i %l %u %t “%r" %s %b “%{Referer}i”
“%{User-Agent}i”'/>
```

From the DXSHELL command line interface on the DX appliance, use the command:

```
dx% set server customipheader rlnclientipaddr
```

to set the custom header field in which the DX appliance will insert the origin client's IP address.

Now, instead of the DX appliance IP address (10.100.1.66) in the example below appearing in the server's access log:

```
10.100.1.66 - - [19/Sep/2001:18:06:52 -0700] “GET / HTTP/1.1” 200 182 “-”
“Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)”
```

the origin client's IP address,192.168.40.247, will be recorded:

```
192.168.40.247 - - [19/Sep/2001:18:06:52 -0700] “GET / HTTP/1.1” 200 182 “-”
“Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)”
```

Configuring Logging with iPlanet

Configure your DX appliance to send the client IP address in a separate HTTP header.

From the DXSHELL, use the command:

```
dx% set server customiplogheader rlnclientipaddr
```

This parameter can also be set using the WebUI from the Network Settings page.

Ensure that the shared object library from your iPlanet/Netscape server installation can be found by the Netscape binary at run time. You can do this by either copying the `rlnLogTrans.so` library into your Netscape distribution's `lib` directory, or by setting the LD_LIBRARY_PATH environment variable to include a directory containing the `rlnLogTrans.so` library.

Next, you will need to edit the `obj.conf file` for your Netscape server's installation. You cannot do this via the Netscape admin server. Find the `obj.conf` file (usually in the "configure" directory for a particular server instance). Open it with your favorite editor and add three lines:

1. At the beginning of the file `obj.conf`, near all the other Init functions (order is not important):

```
Init fn="load-modules" shlib="rlnLogTrans.so" funcs="rlnLogTransInit,rlnLogTrans"
Init fn="rlnLogTransInit" customiplogheader="rlnclientipaddr"
```

The value for customiplogheader can be set to whatever you want to name the custom HTTP header used to pass the client's IP information. If it is not set, it defaults to "`rlnclientipaddr`".

2. Now locate the AddLog line for the flex log. It will look something like:

```
AddLog fn="flex-log" name="access"
```

3. Put in the following line BEFORE the flex-log line:

```
AddLog fn="rlnLogTrans"
```

This forces the `rlnLogTrans` module to run before the logging module.

If you have followed the instructions in the Juniper Administrator's Guide and have converted your "flex-init" line to look for `rlnclientipaddr`, you should switch it back to the stock version. For example, change:

```
Init fn="flex-init" access="/opt/netscape/server4/https-servername/logs/access"
format.access="%Req->headers.rlnclientipaddr% - %Req->vars.auth-u ser% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

back to

```
Init fn="flex-init" access="/opt/netscape/server4/https-servername/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

Save your `obj.conf` file and restart the server.

Now instead of the DX appliance IP address (10.100.55.30) in the example appearing in the server’s access log,

```
10.100.55.30 - - [05/Jan/2004:21:58:53 -0800] "GET / HTTP/1.1" 200 24582
```

you get the client’s IP address:

```
192.168.0.9 - - [05/Jan/2004:21:59:40 -0800] "GET / HTTP/1.1" 200 24582
```

If the DX appliance is removed from the network, you get a log of whatever is upstream, such as a load-balancer or a router.

```
10.100.55.1 - - [05/Jan/2004:22:03:24 -0800] "GET / HTTP/1.1" 200 24582
```

Configuring Logging with NetCache

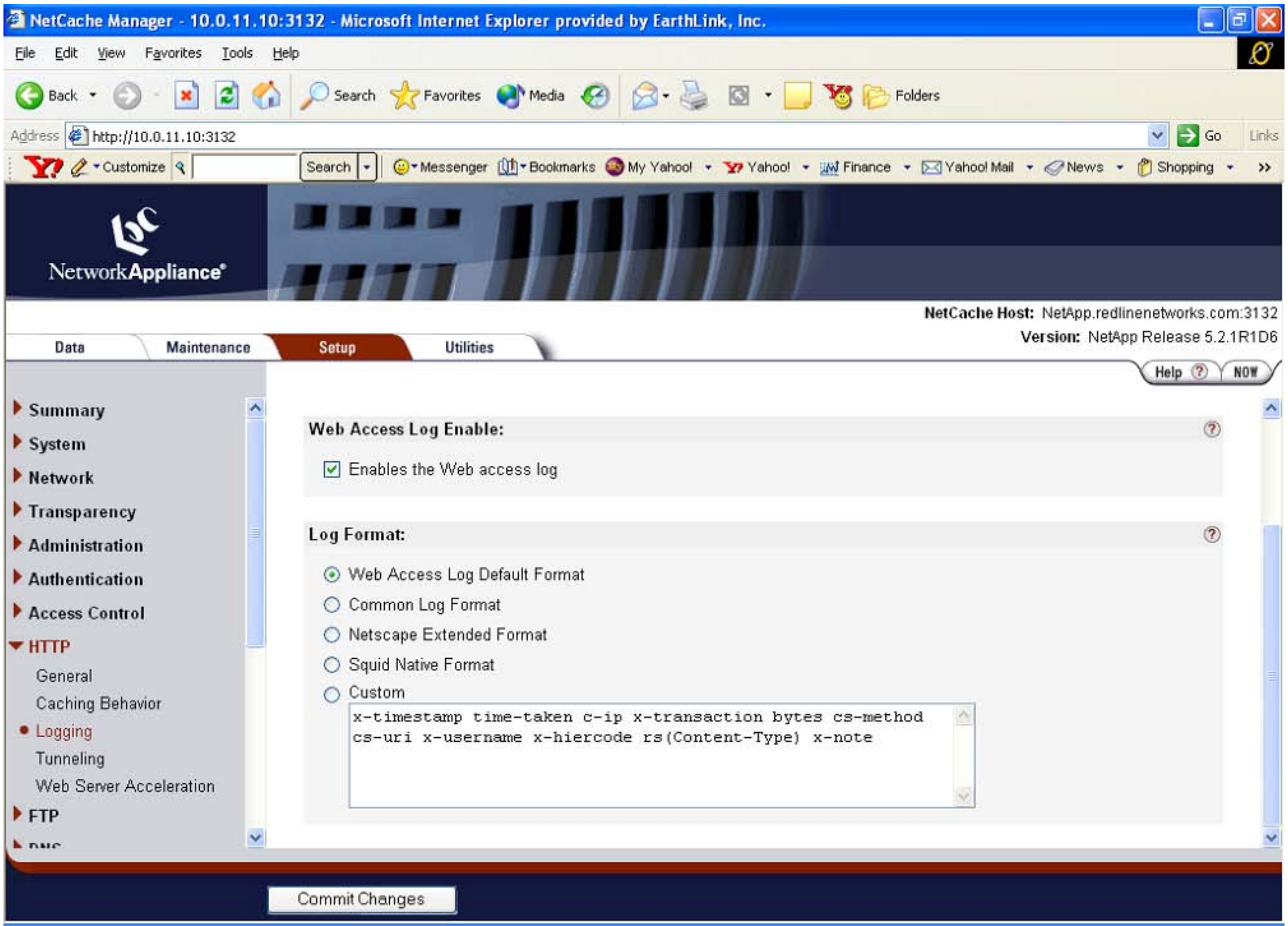
1. From the DXSHELL command-line interface on the DX appliance, type the command:

   `dx%` **`set server customipheader rlnclientipaddr`**
   to set the custom header field in which the DX appliance will insert the origin client's IP address.

2. Configure the NetCache to retrieve the client IP from the custom HTTP header, `rlnclientipaddr`:

   - Log into the NetCache Web Administrator
   - Navigate to the Setup->HTTP->Logging setup screen (refer to Figure 49).

Figure 49: The NetCache Logging Setup Screen



- If the current log format setting is "custom", then skip to the next step. Otherwise, copy the log format variables used by your current log format setting (in the screenshot above)

```
x-timestamp time-taken c-ip x-transaction bytes cs-method cs-uri
x-username x-hiercode rs(Content-Type) x-note)
```

change the Log Format setting to custom and paste the variables into the Custom Log Format setting.

- In the log format variable string, replace:

```
c-ip
```

with

```
cs(rlnclientipaddr)
```

- Save your changes.

- Test your new configuration. Now, instead of the DX appliance IP address appearing in the server's access log, the origin client's IP address will be recorded.

Chapter 13
Server Load Balancing

This chapter describes Server Load Balancing for the DX Application Acceleration Platform, discussing the following topics:

Overview

The DX Application Acceleration Platform can be configured to perform Server Load Balancing between target hosts. Content-intensive applications require more than a single target host to provide adequate processing power, and customers need the flexibility to deploy additional target hosts quickly and transparently to end-users. Server load balancing allows the distribution of service requests across a group of target hosts. Server load balancing also addresses issues important in networks; these are: improved scalability, improved performance, high availability, and disaster recovery. The Server Load balancer uses Layer 4 Switching (L4S) to support various static and dynamic protocols.

While most web-enabled applications and web sites primarily use HTTP, a typical Enterprise site also has DNS services that use the Transmission Control (TCP) or User Datagram (UDP) protocols and web links that involve FTP transfers. (These are used for services such as software updates.) The Juniper DX Application Acceleration Platform provides universal load balancing for HTTP/S, FTP, UDP, and TCP traffic. The DX appliance applies its highly-efficient, fewest outstanding request-based load balancing technology to the HTTP/S protocols, and connection-based load balancing to the remaining protocols.

In order to perform L4S switching and support various static and dynamic protocols, the DX appliances provide support for half- and full-NAT as explained below.

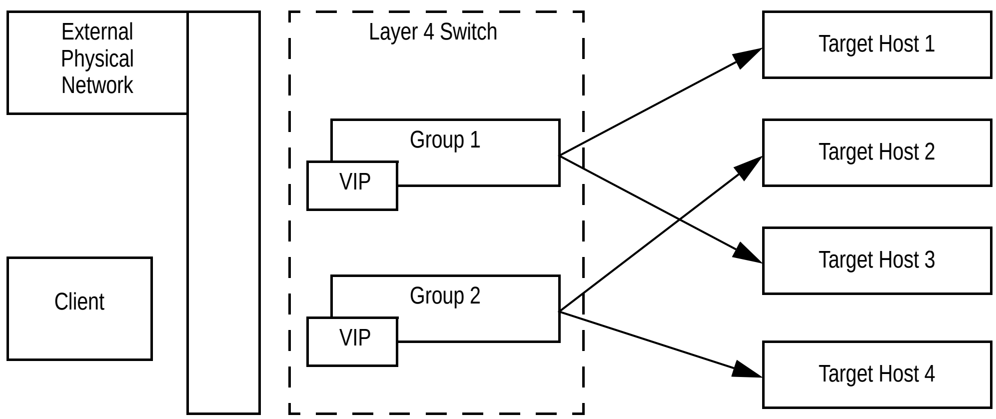
SLB General Operation

The general operation of the L4S configuration is to load balance incoming client connections over FTP, SMTP, and other non-HTTP protocols to the least loaded target host. Traffic flows may undergo full- or half-NAT translations as configured.

SLB Grouping

The L4S switch has a concept of a “group” that is similar to the cluster that exists within the DX appliance. A group represents a collection of target hosts, any one of which is capable of servicing a request. Load balancing rules are then applied to a particular group. Corresponding to each group is a Virtual IP address (VIP) which is aliased on the L4S. Multiple groups can be created. Figure 50 shows what might occur for a physical/logical combination.

Figure 50: Server Load Balancing Groups



There is an external physical network and it refers to the target network into which the L4S is placed. In the example shown in Figure 51, the first L4S Group has two target hosts and the second L4S Group also has two.

SLB Group Health

In order to properly balance traffic between the various target hosts, the L4S must be aware of the health of each target host and remove non-responsive hosts from rotation. The DX appliance tries to establish a TCP connection with each target host. If the connection is successfully made, then the target host is considered operational. If the TCP connection fails, then the target host is considered down.

Port Symmetry

With Port symmetry, the Destination IP is changed (called half-NAT), or both a Destination IP and Source IP (full-NAT) rewrite is performed.

Connection Handling

TCP connections, as made between the client, the L4S, and the target host, are symmetric in nature. The inbound client packets are sent to the L4S and the outbound target host packets are sent via the L4S. The destination IP rewrite, source IP rewrite, or both are determined by the L4S configuration. The mechanism can be illustrated as shown in Figure 51 in full- and half-NAT configurations.

Figure 51: NAT Operation

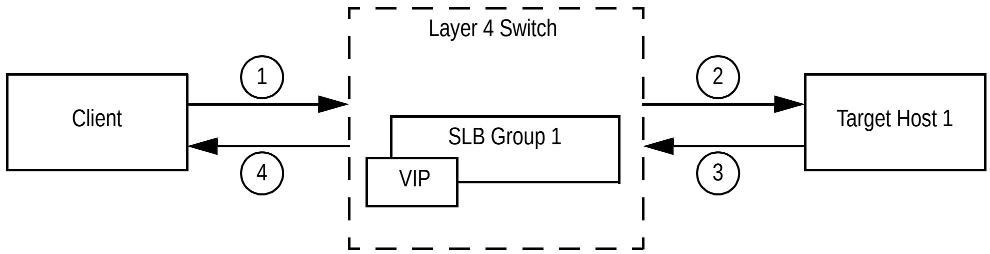


Table 12: Full- and Half-NAT Operation

	Mode			
	Proxy	Full-NAT	Half-NAT	DSR
Notes	Current Mode (No SLB)		The DX Appliance acts as a Gateway for the Target Host	
1	SIP = C DIP = RL_VIP	SIP = C DIP = RL_VIP	SIP = C DIP = RL_VIP	SIP = C DIP = RL_VIP
2	SIP = RL_Real DIP = TH_1	SIP = RL_NAT DIP = TH_1	SIP = C DIP = TH_1	SIP = C DIP = TH_1
3	SIP = TH_1 DIP = RL_Real	SIP = TH_1 DIP = RL_NAT	SIP = TH_1 DIP = C	SIP = TH_1 DIP = C
4	SIP = RL_VIP DIP = C	SIP = RL_VIP DIP = C	SIP = RL_VIP DIP = C	N/A

The L4S switch forwards packets from a client to the appropriate target hosts. The packet forwarding mechanism operates on both new connections and existing ones. For new connections, as identified by a TCP SYN packet, the L4S must determine an appropriate destination. If there is no appropriate destination (as determined by all target hosts for a Group in a non-responsive state), then the packet is dropped. The determination of an appropriate target host where the packet is top be forwarded is determined by the active load balancing scheme as applied to the Group. These balancing schemes are discussed in "Load Balancing Policies" on page 214.

The L4S also monitors the packet flow of each TCP connection to determine when to purge the L4S client connection table entries. In order to resolve a proper TCP teardown, the L4S must know whether it was the client or the server initiated the close. In both full- and half-NAT modes, the FIN and RST packets route through the L4S, and the L4S notes to which TCP session the FIN or RST corresponds. The L4S then forward the TCP session information on to the outbound gateway/router.

Load Balancing Policies

Load balancing policy dictates how the traffic is distributed among all the active servers. The SLB supports the following load balancing policies.

- Round Robin: All the servers in the list are used sequentially for every new TCP session. For example, if there are three servers S1, S2, and S3, the first request goes to S1, the second request goes to S2, and the third request goes to S3. The list wraps around when it reaches the end.

- Least Connection: The server is chosen based upon the number of outstanding active connections to each of the active servers. Once a new request comes in from the client, the server with the least number of active connections is chosen.

- Weighted Round-Robin: The servers are chosen semi-sequentially. A server is chosen based upon its weight. The larger the weight, the higher the probability of the server being chosen. The number of requests per server can be calculated by following equation:

$$Ci = C \times \left(\frac{wi}{\sum w} \right)$$

  where:

  Ci = Connections to Server i

  C = Total number of connections

  wi = Weight of server i

  $\sum w$ = Sum of weights of all servers

  Round-robin can be considered as a special type of "Weighted Round-Robin" where all the servers have equal weight. Alternately a server can be taken out of rotation by assigning a weight of zero.

- Backup Chaining: Whenever a new connection request comes in, the first active target host in the list is picked up. This makes the ordering of the target hosts important. Target hosts have to be added in order of decreasing importance.

  If the primary server goes down, then the next active server in the list is chosen for the subsequent connections. This chaining goes on till the end of the list. The primary is always the first target server that the user configured, and all other target servers (backup 1, 2, . . . etc.) in the list are backups. The list will wrap when it reaches the last server in the list.

  You must also specify whether you want to revert back to the original target server as soon as that server becomes available after a failure, or revert to the original server only when the backup server goes down.

- Maximum Connections: Maximum connections (maxconn) is a load balancing policy where the first server in the list will be given the first N number of concurrent connections, and will then forward the next N concurrent connections to the next server in the list (and so on). For example, if Target Host 1 is configured to have a maxconn value of 300, all the first 300 sessions will be forwarded to Target Host 1 and any new connections after these will go to Target Host 2. If Target Host 2 is sequentially configured to have 500 connections, the next 500 connections are forwarded to Target Host 2. The range for the maxconn value is from one to 2000, and the default value is 200.

  This behavior is chained throughout all the active target hosts present. However, if the client "sticky" feature is enabled, then the sticky entries will take precedence over the load balancing algorithm.

- Weighted Least Connections: Weighted Least Connections is a load balancing policy that distributes traffic to various target servers based on the weights assigned to each of the target servers. The load distribution is asymmetrical owing to difference in the weights of the server. The "Weighted Least Connection" algorithm works similar to Least Connection algorithm. In fact, the Least Connection policy can be thought of as special case of Weighted Least Connections where all the target servers have weight of 1.The next server to be selected to forward the request is derived with the formula as:

  $$Si = min\left(\frac{C1}{W1}, \frac{C2}{W2}, \ldots, \frac{Cn}{Wn}\right)$$

  where:

  Si = The next server to be given the new connection

  Ci = The current number of established sessions to the server

  Ci = The weight of the server

  $\frac{Ci}{Wi}$ = The effective number of connections to the server

  For example, a server with 200 outstanding connections and weight of 2 is same as a server with 100 outstanding connections and weight of 1. A weight of 0 takes a server out of rotation, but the health checking will be still on.

Failover

The L4S is capable of failing over to another L4S using a failover mechanism similar to the current failover mechanism. The active L4S sends RMMP health messages that the other L4S receives. If a certain number of health messages are not received within a time window, the second L4S takes over processing new requests.

For additional information, see "Configuring for High Availability" on page 113.

SLB Configuration Commands

The following commands are used to configure the L4S NAT support. Note that all the commands except the `show` and `clear stats` commands take effect only after a `write` operation. The syn flood protection is always enabled.

Add Commands

To add a new group with optional name and VIP and port:

```
dx% add slb group [name] < vip:port >
```

Delete Commands

To delete a group specified by name. Using `all` will delete all groups:

```
dx% delete slb group < name | all >
```

Set Commands

To start and stop the Server Load Balancer, type the command:

```
dx% set slb < enabled | disabled* >
```

The default is disabled.

To set a new target host with a “real” IP address, type the command:

```
dx% set slb group [name] targethost < ip:port >
```

To set a protocol for the switch group, type the command:

```
dx% set slb group <name> protocol <tcp* | udp>
```

The default is tcp.

To set a load balancing policy for a defined group, type the command:

```
dx% set slb group <name> policy <roundrobin* | leastconn | bkupchain [norevert* |
revert] | weightedrr | maxconn | weightedlc>
```

The round-robin policy is the default. For details on the various policies, see “Load Balancing Policies” on page 214.

To set the maximum number of connections per targethost when the maxconn load balancing policy is in effect, type the command:

```
dx% set slb group < name | all > targethost <ip:port | all > maxconn <number>
```

The range for the maxconn value is from one to 2000, and the default value is 200. This value is effective only when the maxconn load balancing policy chosen.

To set a full or half NAT policy for the switch group, type the command:

```
dx% set slb group <name> nat <half | full*>
```

The default is full.

Since the L4S switch can operate in DSR mode, it may not see the packets going from target host to the client. This makes it difficult for the L4 switch to track the connection state, so it uses a timer to purge the sessions.

To set the ports from an SLB group that can be NAT-ed, type the command:

```
dx% set slb group <name> nat port <start | end> <value>
```

There is no default value. The start and end port numbers are inclusive.

To enable “stickiness” of a particular client to a server within a group, type the command:

```
dx% set slb group < name > sticky < enabled | disabled* >
```

Stickiness results in a client always being connected to the same server (if reconnected before the timeout). The default setting is disabled.

To set the timeout for the “stickiness” of a particular client to a server, type the command:

```
dx% set slb sticky timeout <minutes>
```

The default value is 120 minutes.

To set the timeouts for the active sessions, type the command:

```
dx% set slb session timeout active <secs>
```

The default value is 100 seconds.

A closewait is a session that is waiting to be closed, but has not closed as of yet. To set the timeouts for the close wait sessions, type the command:

```
dx% set slb session timeout closewait <secs>
```

The default value is 15 seconds.

A synwait is a session with a three-way handshake not terminated (SYN sent by client and waiting for a SYN/ACK from the server or SYN sent by client and SYN/ACK sent by the server, but waiting for an ACK from the client). To set the timeouts for the synwait sessions, type the command:

```
dx% set slb session timeout synwait <secs>
```

The default value is 10 seconds.

When active sessions are purged, a reset can be sent to the client and server to indicate that the connection has been terminated. To enable or disable the sending of resets to the client, type the command:

```
dx% set slb advanced reset client <enabled* | disabled>
```

The default value is enabled.

To enable or disable sending of reset to the target host, type the command:

```
dx% set slb advanced reset server <enabled* | disabled>
```

The default value is enabled.

Health Check Commands

Periodic health checks of the target servers are conducted to check their status. The following commands set the parameters associated with the health checking. Health check is a default feature and cannot be turned-off.

To set the health check interval when the target hosts are up, type the command:

```
dx% set slb healthcheck interval up <secs>
```

The default value is 45 seconds.

To set the health check interval when the target hosts are down, type the command:

```
dx% set slb healthcheck interval down <secs>
```

The default value is 20 seconds.

To set the health check interval for TCP SYN, type the command:

```
dx% set slb healthcheck interval syn <secs>
```

The default value is 10 seconds.

To set the maximum number of health check tries before giving up, type the command:

```
dx% set slb healthcheck maxtries <number>
```

The default value is three tries.

Failover Commands

To enable or disable the “forcemaster,” type the command:

```
dx% set slb failover forcemaster <enabled | disabled*>
```

Enabling forcemaster allows a switch to snatch the active status from another switch of higher node-id. The default value is `disabled`.

To set the multicast address for the failover mechanism, type the command:

```
dx% set slb failover mcastaddr <ip addr>
```

The default value is 227.0.0.6.

To set the bind address for the failover mechanism, type the command:

```
dx% set slb failover bindaddr <ip addr>
```

The bindaddr is the address that you want to use to send Redundancy Multicast Messaging Protocol packets (for RMMP failover). The default value is Null, which indicates that the SLB should use the interface address as the bind address.

To set the nodeID of the SLB failover unit, type the command:

```
dx% set slb failover nodeid <number | auto*>
```

Setting nodeID to auto results in the node-id being generated automatically. The default is auto.

To set the port for failover communication, type the command:

```
dx% set slb failover port peer <port>
```

The default is 9200.

To enable or disable the use of Virtual MAC on the interface, type the command:

```
dx% set slb failover vmac < enabled | disabled* >
```

The default setting is disabled.

To set the Virtual Router ID of the failover unit, type the command:

```
dx% set slb failover vmac id < id >
```

The ID is a number between 1 and 255, both inclusive.

Clear Commands

To clear the statistics for the group, type the command:

```
dx% clear slb group <name | all> stats
```

To clear the target host statistics, type the command:

```
dx% clear slb group targethost <ip:port> stats
```

To remove a target host specified by index, type the command:

```
dx% clear slb group <name> targethost < ip:port | all >
```

Using all will delete all groups.

To clear overall statistics, type the command:

```
dx% clear slb stats
```

Show Commands

To display the basic L4S configuration parameters, type the command:

```
dx% show slb
```

Table 13 shows the possibilities.

:

Table 13: Show SLB Command Permutations

Switch Status	Meaning
Disabled	The Server Load Balancer is off
Enabled (stand-alone)	The SLB is in stand-alone mode
Enabled (active)	The SLB is enabled for failover and is the active switch
Enabled (passive)	The SLB is enabled for failover and is the backup switch

To display the group characteristics, type the command:

```
dx% show slb group <name | all>
```

You should see a listing like this:

```
dx% show slb group 1
========================================
group 1
vip: 192.168.15.62
port: 22
policy: bkup chain <=====
protocol: tcp
nat: full
nat port start: 1024
nat port end: 8000
sticky: disabled
========================================
```

To display the SLB state for a server/switch, type the command:

```
dx% show slb status
```

To display the group statistics, type the command:

```
dx% show slb group <name | all> stats
```

To display the overall statistics for the switch, type the command:

```
dx% show slb stats
```

To display the blade statistics, type the command:

```
dx% show slb targethost <ip | all> stats
```

Using all will display all blades.

To display the failover status, type the command:

```
dx% show slb failover
```

Configuring Server Load Balancing

This section describes the procedure to program the DX appliance Server Load Balancer. For a bare minimum setup, all you need to do is to add a group with a VIP:Port address, and set a target host in that group.

Adding a Group

The group that you will be adding is a listen VIP for an application or protocol being load balanced. It creates an IP:Port pair to listen to the incoming traffic.

Add a group by typing the command:

```
dx% add slb group [name] ip:port
```

The name of the group is optional.

Adding a Target Host

A target host is a single server that is part of a group.

Add a target host to a group by typing the command:

```
dx% set slb group <name> targethost <ip:port>
```

Setting the Group Parameters

To configure a group, you will need to set up these parameters:

- Network Address Translation (NAT) Parameters
- Port Range Start and End
- Group Protocol

The default values are:

- NAT: full
- NAT Port Start: 1024
- NAT Port End: 8000
- Protocol: tcp
- Policy: Least Connection

Setting Network Address Translation (NAT) parameters

With Full-Nat, Network Address Translation is performed on both the source address and the destination address. The interface IP is used as the source IP address to connect to the target host. There are no modifications required to the target hosts. This is the default setting.

With Half-Nat, the source address is retained, and Nat is only performed on the destination address. For this configuration to work, the target host should have the default route pointing to the SLB.

Set NAT to Full-Nat by typing the command:

```
dx% set slb group <name> nat full
```

or

Set NAT to Half-Nat by typing the command:

```
dx% set slb group <name> nat half
```

Setting the port range start

This command is effective only when the NAT is set to `full` The start port denotes the starting port for the NAT. The value should be between 0 - 65535.

Set the port range start by typing the command:

```
dx% set slb group <name> nat port start <number>
```

Set the port range end by typing the command:

```
dx% set slb group <name> nat port end <number>
```

This command is effective only when the NAT is set to `full`. The end port denotes the ending port for the NAT. The value should between 0 - 65535 and should be more than the start port.

Setting the group protocol

This command sets the protocol for the group to either TCP or UDP.

Set the group protocol by typing the command:

```
dx% set slb group <name> protocol < udp | tcp >
```

Setting the group load balancing policy

This command sets the load balancing policy for the group.

Set the group load balancing policy by typing the command:

```
dx% set slb group <name> policy < leastconn | roundrobin | bkupchain | weightedrr |
maxconn>
```

Refer to “Load Balancing Policies” on page 214 for a description of the load balancing options.

Deleting a Group

The delete command deletes a group that was previously added. Using the keyword `all` deletes all of the groups. When deleting a group, all the servers that were added under the group are also deleted.

Delete a group by typing the command:

```
dx% delete slb group < name | all >
```

Deleting a Server from a Group

Delete a server from a group by typing the command:

```
dx% clear slb group < name > targethost < targetip | all >
```

Statistics

These commands are used to view statistics for the Server Load Balancer.

Overall Statistics

Display the overall statistics by typing the command:

```
dx% show slb stats
```

Clear the overall statistics by typing the command:

```
dx% clear slb stats
```

Group Statistics

Display the group statistics by typing the command:

```
dx% show slb group < name | all > stats
```

Clear the group statistics by typing the command:

```
dx% clear slb group < name | all > stats
```

Target Host Statistics

Display the target host statistics by typing the command:

```
dx% show slb group < name > targethost < serverip:port | all > stats
```

Clear the target host statistics by typing the command:

```
dx% clear slb group < name > targethost < serverip:port | all > stats
```

Client IP Sticky

Stickiness results in a client always being connected to the same server (if reconnected before the timeout).

Set client IP stickiness by typing the command:

```
dx% set slb group < name | all > sticky enabled | disabled
```

The default setting is disabled.

Set the sticky timeout by typing the command:

```
dx% set slb sticky timeout < minutes >
```

Failover

Enabled failover by typing the command:

```
dx% set slb failover enable
```

Set VMAC by typing the command:

```
dx% set slb failover vmac < enabled | disabled >
```

Set the VMAC ID by typing the command:

```
dx% set slb failover vmac id < id >
```

Chapter 14
Global Server Load Balancing

This chapter describes Global Server Load Balancing for the DX Application Acceleration Platform, discussing the following topics:

Overview

Global Server Load Balancing (GSLB) allows installations with multiple sites the capability of continuing to operate when one (or more) of their sites goes down. GLSB automatically takes that site out of rotation until it is available again. It has the added benefit that it can dynamically load balance across several sites, routing more traffic to the faster sites.

Installations where this will be of concern are those with multiple replicated sites, generally distributed across different physical data center locations. For instance, you might have data centers in New York, Chicago, and Seattle, and want to seamlessly shift requests among the centers. If a center goes down, requests should be automatically shifted to the remaining centers without any intervention on your part. When the center returns to service, it is re-entered into the mix.

NOTE: GSLB is an optional feature that requires a license. Contact your Juniper Sales Representative to obtain a license.

GSLB is implemented by manipulation of DNS records. Remote clients contact the site's DNS server asking for a hostname, and receive a response containing a list of IP addresses that correspond to servers for them to contact. The clients then attempt to contact the first host on the list. If that host does not reply, the client then tries the next, continuing in this fashion until either a successful contact has been made or the client reaches the end of the list.

By manipulating the order of the addresses in the list, a form of load balancing can be achieved. In the simplest case, the DNS server knows nothing about the state of the hosts its serving and simply follows a pre-determined algorithm, commonly just round-robin rotating among a pre-defined set of hosts. However, by adding a mechanism by which the DNS server can check the health of the hosts, this can be extended to perform some advanced load balancing, and even be used to cleanly remove hosts from service. This is an advantage for both known events such as scheduled downtime, or for unplanned events such as a network outage.

DNS Proxy Filter

The DNS-proxy filter listens on a VIP for DNS requests and forwards them intact to the real DNS server, similar to the manner in which the server forwards requests from a listen VIP to a target host. In the case of an external DNS server, this is simply the IP and port of the target server. If you elect to use the internal DNS server (see DNS Server on page 232), the target server becomes loopback, and BIND is started, listening on loopback.

The blade listens for both TCP and UDP requests and forwards them on to the real DNS server. For instance, if the client makes a UDP request, the filter forwards that to the DNS server as a UDP request.

Requests going to the DNS server are inspected. If the request is a name-lookup request, a further check is done to see if the name is one of the configured GSLB groups. If the request does not meet both of these criteria, the request is forwarded intact. Responses from the DNS server are passed directly on to the client. Note that there is no caching of DNS responses. This is not a caching DNS proxy!

If the request is determined to be a lookup from one of the GSLB groups, the filter composes the response directly based on the current health status of members in the GSLB group and the load-balancing algorithm selected.

If the request is a UDP request, the filter responds with a UDP response, and if it is a TCP request, it will respond with a TCP response. The filter conforms to the specification in RFC 1035 with regards to long responses. This specification states that a UDP response larger than 512 bytes shall simply be truncated, and a truncation bit shall be set in the DNS packet header. This indicates to the client that it should retry the request as a TCP one. If the client chooses to make a second request, this is considered to be a completely separate transaction, and the new response may be slightly different, depending on the group members' health and the progression of the load-balancing algorithm. Also note that even TCP DNS requests have a maximum length of 65535 bytes; this is imposed by the DNS protocol itself.

Group Member Health Checking and Load Balancing

If a group's load-balancing policy is set to "forward," all queries for that group are forwarded onto the real DNS server, regardless of the member's health state. If the group's policy is set to "fixed," "random," or "roundrobin," the member's health is checked and then a load-balancing algorithm is applied.

Health checking for group members is limited to ping checks. Group members are pinged asynchronously from DNS requests. A member is pinged once a second, and three seconds are given for the member to reply. No pings are sent to that member in the interim. Once the three second response time expires, the member

is pinged again, and given another three seconds to reply. If no reply is received, a third ping is sent. If the member still fails to respond within three seconds, the member is considered to be down. Note that this means that it can take up to nine seconds to determine that a member is unavailable.

Members that appear in more than one group have their health checking coalesced into one probe per health-checking round. For example, even if a member IP appears in five groups, it will still only be pinged once per second; not five times.

If a member is considered to be down, it is still sent a ping at one-second intervals. If it resumes responding to pings, it is immediately considered to be available.

Once all members of a group have had their availability checked, the group's load-balancing algorithm is used to determine the final ordering of members in the DNS response. If the group's policy is "fixed," no load-balancing is done, and the hosts are returned in the order they were added to the group. If the group's policy is "random," the ordering will be random. If the ordering is "roundrobin," the hosts are returned round-robin. If the group's policy is "forward" the request is forwarded on regardless of the member's health. Note that in the last case members are still pinged even though the results are unused. At startup time, all hosts are considered to be down until responses to pings are received. Therefore it may take a few seconds for hosts to appear in DNS queries.

Statistics

There are two levels of statistics available: filter-level and group level. The filter-level statistics reports the DNS queries the filter has handled. See Table 14.

Table 14: GSLB Statistics

Statistic	Description
Requests	All complete DNS requests
Replies	Replies originating from the filter itself
Forwards	Requests forwarded onto the DNS server
Replies from DNS server	Replies from the DNS server
Errors	The count of DNS packet-parsing errors (malformed or otherwise invalid requests)

Group-level statistics display the availability of a group member as determined from ping checks.

Deployment

Deployment of GSLB should be quite simple in most cases. Your site does not need to change its DNS server configuration at all. However, since DNS server IP addresses are maintained by external domain registrars, you will have to make some minor changes in order to make the DX appliance the primary DNS address.

This can be effected by one of two means. Either set the DX appliance’s DNS filter listen address to a new IP address and put in a request to your registrar to update your DNS record to point to the DX appliance, or set the DX appliance's DNS filter listen address to the current DNS server's IP address, and hide the current DNS server behind a Network Address Translation (NAT) device.

Once this is complete, you need to define the DX appliance's GSLB hosts. Although the host IP addresses must be publicly available, the health-checking IP addresses do not. Because the configuration allows for the health-checking addresses to be defined independently from the actual IP addresses published by DNS, the health-checking IP's can be located on a private back-channel.

GSLB Configuration Commands

Basic DNS Filter Configuration Commands

The basic `GSLB DNS` filter configuration commands are described here. Commands that affect the statistics and health gathering are described in “DNS Filter Configuration Commands” on page 229.

All of the commands for configuring the GSLB feature are grouped under the top-level “gslb” keyword. For instance, the set command for this feature consists of three sub-commands:

- `set gslb filter`
- `set gslb group`
- `set gslb dns`

Note that the `set gslb dns` command is different from the `set dns` command that sets the administrative DNS for the DX appliance, and is not necessarily the same as the DNS used for GSLB.

Set Commands

To set the filter's Virtual IP (VIP) address for listening to public DNS requests, type the command:

```
dx% set gslb filter listen vip <IP>
```

To set the filter's listen port, type the command:

```
dx% set gslb filter listen port <N>
```

This is limited to the standard port limit of 65536, and defaults to the standard DNS port 53.

To set the IP address of the DNS server that the filter is to contact, type the command:

```
dx% set gslb filter target ip <ip|internal>
```

If the value consists of the keyword “internal,” the filter ignores the port setting (see below) and contacts the internal DNS server based on its configured IP and port.

To set the port for the DNS server that the GSLB filter is to contact, type the command:

```
dx% set gslb filter target port <N>
```

The default is the standard DNS port 53. If the filter's target IP is set to the keyword “internal,” this value is ignored.

The GSLB filtering service is started and stopped using the usual service up and/or down commands:

To start the GSLB filtering service, type the command:

```
dx% set gslb up
```

To stop the GSLB filtering service, type the command:

```
dx% set gslb down
```

Show Commands

There is a single show command that displays all of the configuration options available to the DNS filter:

```
dx% show gslb filter
```

This shows all the values set using the `set` commands.

DNS Filter Configuration Commands

Add Commands

To create a GSLB group, type the command:

```
dx% add gslb group [name]
```

The name argument is optional. If no name is specified, one will be created for the GSLB group. The group’s name is strictly for identification purposes, and can be changed with a set command.

Set Commands

To change the name of a GSLB group, type the command:

```
dx% set gslb group name <name>
```

To set the host name for the group, type the command:

```
dx% set gslb group <name> hostname <FQDN>
```

To add a host (group member) to the GSLB group, type the command:

```
dx% set gslb group <name> member <IP>
```

This adds a host to the list of hosts to be considered for GSLB health-checking and load balancing.

To set the group's load balancing policy, type the command:

```
dx% set gslb group <name> policy <policy>
```

The policy must be one of <roundrobin | fixed | random | forward > , and defaults to "roundrobin."

To set the Time to Live (TTL) for the group, type the command:

```
dx% set gslb group <name> ttl <seconds>
```

The TTL must be a value between 300 and 4294967295 seconds (inclusive), and defaults to. 300.

The `<authdomainname>` and `<authservername>` parameters are optional, but may be set as well. To set the group's `<authdomainname>`, type the command:

```
dx% set gslb group <name> authdomainname <FQDN>
```

To set the group's `<authservername>`, type the command:

```
dx% set gslb group <name> authservername <FQHN>
```

Where `<FQDN>` is the Fully-Qualified Domain Name, and `<FQHN>` is the Fully-Qualified Host Name.

Clear and Delete Commands

To remove a host from the GSLB group, type the command:

```
dx% clear gslb group <name> member <ip | all>
```

To clear a group's host name, type the command:

```
dx% clear gslb group <name> hostname
```

To clear a group's `<authdomainname>`, type the command:

```
dx% clear gslb group <name> authdomainname
```

To clear a group's `<authservername>`, type the command:

```
dx% clear gslb group <name> authservername
```

The TTL is not clearable; it may only be set to a different value.

The group name is also not clearable. The entire group may be deleted with a delete command. Note that this is allowable even if the group still has members -- all the members will be deleted with the group.

To delete an entire group, type the command:

```
dx% delete gslb group <name>
```

Show Commands

To show the GSLB filter statistics, type the command:

```
dx% show gslb filter stats
```

The information displayed is described in “Statistics” on page 227.

To show all of the configuration options available for a DNS filter host, type the command:

```
dx% show gslb host <name | all>
```

When this command is executed either without a specified groupname or with the keyword “all”, the configuration information for all groups will be displayed.

Group member information is also available through show commands. To show all of the settable group parameters, type the command:

```
dx% show gslb group <name> hostname
```

When this command is executed without a specified groupname or with the keyword “all”, the configuration information for all the groups will be displayed.

To show the group member health information, type the command:

```
dx% show gslb group <name> status
```

This shows the group member health information, as determined from ping checks.

DNS Server

The DNS Server feature provides basic Domain Name System (DNS) server functionality for the DX Application Acceleration Platform. The DNS server is used together with Global Server Load Balancing (GSLB), and cannot be used as a standalone name server. This is a convenience feature, not a fully-configurable server, and is intended to be used when you need rapid deployment of an easily available name server as opposed to a complete set of DNS services.

The DNS server can be configured to be an authoritative primary (Master) server for a particular domain. The BIND's named will be used as the DNS server. A subset of named's configuration is exposed through DXSHELL. The DNS server can only be used by the GSLB Proxy and cannot be used as a standalone name server. When GSLB is being used, the DX appliance acts as a proxy to DNS servers.

With the DNS server functionality, the DX appliance can be used as the DNS server also, eliminating the need for external DNS servers.

Configuring the DNS Server

Add Commands

To add a domain to the DNS Server, type the command:

```
dx% add gslb dns domain <domain>
```

This adds a start of authority record for the specified domain.

Set Commands

To add a name server record for a sub-domain in the specified domain, type the command:

```
dx% set gslb dns domain <domain> ns <subdomain> <server name>
```

When “@” is specified as the subdomain, the name server record is added for the domain itself. This means that any sub-domain request (i.e., mail.domain.com, sales.domain.com, service.domain.com, etc.) will automatically be directed to that server. When the sub-domain or server name does not end in a period (.) (is not fully qualified), the name server appends the domain name to the subdomain when responding to queries. There can be multiple name server records for a subdomain.

To add an address record for a host in the domain, type the command:

```
dx% set gslb dns domain <domain> a <host> <ip>
```

When the host parameter does not end in a period (.), (is not fully qualified), the name server appends the domain name to it when responding to queries. There can be only one address record for a host in a domain, however, you can have multiple aliases.

To add an alias for a host (e.g., adding a canonical name record for alias) in the domain, type the command:

```
dx% set gslb dns domain <domain> cname <host> <alias>
```

The host must be one of the hosts for which an address record is already configured. If either of host or alias does not end in a period (.), (is not fully qualified), the name server appends the domain name to it when responding to queries. There can be multiple aliases for a host in a domain. For example:

```
dx% set gslb dns domain a.com cname www ftp
dx% set gslb dns domain a.com cname www gopher
```

To add a pointer record (for reverse DNS lookup) for an IP in the specified domain, type the command:

```
dx% set gslb dns domain <domain> ptr <ip> <host>
```

If the host parameter does not end in a period (.) (is not fully qualified), the name server appends the domain name to it when responding to queries. There can be only one pointer record for an IP in a domain.

To add a mail exchange record for sub-domain in domain, type the command:

```
dx% set gslb dns domain <domain> mx @ <mail server> <priority>
```

A mail exchange record specifies the name of the mail server for the domain. When the sub-domain or mail server not end in a period (.) (not fully qualified), the name server appends the domain name to these when responding to queries. There can be multiple mail exchange records for a subdomain in a domain with different priorities. Priority is a positive integer with zero being the highest priority.

To set the Time to Live (TTL) for the specified domain, type the command:

```
dx% set gslb dns domain <domain> ttl <secs>
```

TTL) allows the administrator to configure how long a DNS record will be cached before it needs to be removed. The default "Time to Live" is 300 seconds. This TTL is used for all the Resource Records in a domain.

To set the contact email for the domain, type the command:

```
dx% set gslb dns domain <domain> contact <email>
```

The contact email is not used by the name server, but is returned on request by DNS clients. The clients can then contact the administrator using this E-mail address, should a need arise. The format for contact E-mail is "name@domain" with the "@" replaced by a dot. So it is "name.domain". The default E-mail is "juniper.$hostname".

To set the serial number for the domain, type the command:

```
dx% set gslb dns domain <domain> serial <N>
```

At the top level of a domain, the name database must contain a Start of Authority (SOA) record. This SOA record contains the current version of the DNS database, various other parameters that define how the DNS server responds for a particular domain, and a serial number. Each time that the information in the SOA record changes, the serial number increments, which allows requestors a quick tool for determining whether changes have occurred.

When a secondary nameserver for the domain contacts the primary nameserver to check if there has been a change to the primary's DNS database, and if the secondary should do a zone transfer, it compares its own serial number against that of the primary nameserver. If the serial number of the secondary nameserver is higher than that of the primary, a zone transfer does not occur. If the serial number of the primary nameserver is a higher number, the secondary nameserver performs a zone transfer and updates its own DNS database.

However, each time that a domain is added to the DNS server, the serial number is set to 1, which can cause confusion. This command allows you to set the serial number.

Deleting Domains and Resource Records

To remove the domain and all its records, type the command:

```
dx% delete gslb dns domain <domain>
```

To delete the specified name server record, type the command:

```
dx% clear gslb dns domain <domain> ns <subdomain> <name server>
```

To delete the specified address record, type the command:

```
dx% clear gslb dns domain <domain> a <host>
```

To delete the specified canonical name record, type the command:

```
dx% clear gslb dns domain <domain> cname <host> <alias>
```

To delete the specified pointer record, type the command:

```
dx% clear gslb dns domain <domain> ptr <ip>
```

To delete the specified mail exchange record, type the command:

```
dx% clear gslb dns domain <domain> mx @ <mail server>
```

Showing the DNS Server Configuration

To display all the Resource Records for all the domains or only for `<domain>`, if specified, type the command:

```
dx% show gslb dns domain [domain | all]
```

Chapter 15

3G Cache

This chapter describes 3G Cache for the DX Application Acceleration Platform, discussing the following topics:

Overview

This chapter describes the caching functionality implemented in the DX Application Acceleration Platform and the associated DXSHELL enhancements. Caching stores frequently requested content in memory on the DX appliance (in-memory cache) to provide improved response times and reduced network bandwidth usage for subsequent requests for the same content.

During an initial request for content (or an object), the content is requested from the origin server, read off of the server's hard-disk, and served to the client. At the same time, it is stored in the cache. Subsequent requests retrieve the content directly from the cache. Caching these requests in the DX appliance's memory can greatly accelerate these transactions while at the same time reducing server load and network bandwidth.

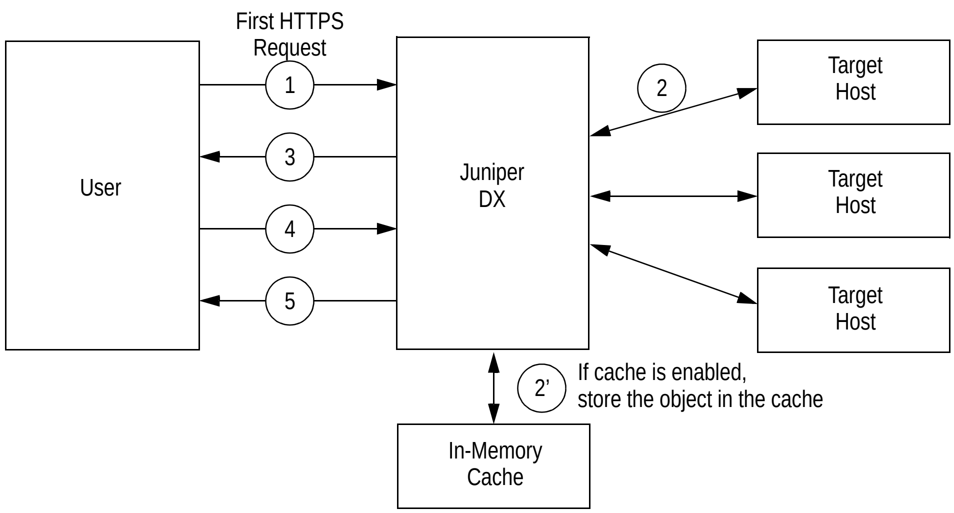
The 3G Cache feature is a licensable item, and requires having an activation key in your license file. If you are upgrading from Release 3.3 and need to use 3G Cache, you must request a new license key from the Juniper Technical Support site (see "The License Key" on page 47). Note that 3G Cache is independent of Overdrive AppRules licensing, and that AppRules are needed to populate and retrieve objects from the cache.

The Juniper Solution

The DX appliance can provide in-memory cache in reverse-proxy mode, servicing requests from clients for a large number of target hosts. The caches are typically deployed to achieve either content acceleration or to help off-load the server. When used for server off-load, the DX appliance (and cache) is in transparent reverse-proxy mode.

In this mode, a request for content to the origin servers (1) is made to a Virtual IP Address (VIP) on the DX appliance (refer to Figure 52). Once the DX appliance receives the server response content (2), it simultaneously delivers the content to the client (3), and also stores the content in its cache/storage (2'). For each subsequent request (4), the response (5) comes from the cache in the DX appliance.

Figure 52: Cache Request Flow



Cache Usage Scenarios

Caching may be used in different scenarios. Some typical ones are:

- Objects are retrieved from in-memory cache - a cache-hit
- Objects are present in the memory but stale - a cache miss
- Objects are not present in the in-memory cache - a cache miss

These conditions are described in Table 15.

Table 15: Cache Usage Conditions

Cache Status	Object Present	Object Absent
Cache Enabled	Cache-Hit	Cache-Miss
Cache Disabled	N/A	Normal DX Operation

Caching Features

The DX appliance supports the following high-level caching features.

Caching and Cache Management

Caching and cache management work in reverse-proxy mode. They allow:

- Objects to be cached in the memory
- Caches to be configured independently of clusters. They are assigned to clusters and enabled or disabled in a manner similar to that of target hosts. Several clusters may use the same cache.
- AppRules to define which objects are to be cached within a particular cluster. A caching AppRule has no effect if a cluster's cache is disabled.

Cache Persistence

AppRules are used to specify the lifetime of the cached objects (i.e., how long to save objects in-memory).

Cache Storage

The cache storage is in-memory, and this implementation provides end users with high performance and great reliability.

Transparency

Clients to the DX appliance are not aware that the DX appliance is caching objects. The flexible controls of the AppRule framework allow administrators to use caching with applications that are incompatible with typical general-purpose caches.

Cache Load Balancing

Cache load balancing is unnecessary since the cache is in-memory only.

Cache Statistics

The following classes of statistics are provided:

- Cache Operational Statistics: Memory usage and other relevant data necessary to monitor the "health" of a cache
- Cache Content Statistics: Object sizes and hitcounts
- Cluster cache-usage: HTTP and I/O statistics similar to target host stats
- Cluster AppRule Stats: Cluster statistics with caching AppRule usage

Cache Placement and Expiration Policy

The AppRules are used to specify which objects to cache, and for how long. Refer to "Show Cluster Cache Commands" on page 242 for additional information.

Multi-Encoding

The cache is capable of not only storing objects in their native format (i.e., HTML, a text document, etc.), but also "derived" formats as well. In particular, it has the ability to store objects in the cache that have been compressed and processed by Page Translator Content rules. These derived formats are called "encodings." This allows for higher throughput because the effort to repeatedly produce a compressed version of a cacheable object is no longer required.

Internally, this means that a single cache entry may actually be stored in multiple encodings. Because of this, a single entry may take up more room in the cache than its literal byte count may imply. For example, if a 30K page is stored in the cache in its native, uncompressed format as well as in its compressed encoding format, there is more than 30K of the cache's memory consumed. However, if clients only ask for a derived format, then only that format is stored. This means that if all browsers to make requests through a DX appliance support compressed documents, then only compressed documents will be stored in the cache. This has the effect of using less space in the cache than would otherwise be required if the document was stored in its native format.

Configuration

3G Cache Commands

The following commands are used from DXSHELL to support in-memory caching:

Add Cache Commands

To add a cache, use the command:

```
dx% add cache <name>
```

For example:

```
dx% add cache secureImages-01_01
(*) dx%
```

The name can be up to 32 characters long, can be any valid character string, and may be integer-only. The valid characters are:

@;$^&*() = ! < > ,[]/_. + -0123456789

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.

Reserved DXSHELL keywords such as "all", "none", and "question" are considered invalid. If a name is not specified, one is automatically assigned. In general, this command conforms to the same rules as "add cluster." Refer to Chapter 1 in the *Command Line Reference* manual for more information.

Set Cache Parameter Commands

To set the total number of objects that can be stored in the named cache, use the command:

```
dx% set cache <name> max_objects <number>
```

The minimum number is 1000 and the maximum is 32000. The default value is 8192.

For example:

```
dx% set cache secureImages max_objects 28000
(*) dx%
```

As a convenience, the number may be abbreviated with a "k" suffix to indicate 1000 objects:

```
dx% set cache secureImages max_objects 2k
(*) dx%
```

To set the base size (in bytes) of the named cache, use the command:

```
dx% set cache <name> size <number>
```

The minimum number is 1,048,576 (1 megabytes) and the maximum is 104,857,600 (100 megabytes). The default value is 10,485,760 (10 megabytes). The actual size of the cache can be somewhat larger than this. As a short cut, the command is:

```
dx% set cache secureImages size 104857601
(*) dx%
```

and can be abbreviated with an "m" suffix to indicate a megabyte (1,048,576 bytes):

```
dx% set cache secureImages max_objects 3m
(*) dx%
```

Clearing Cache Statistics and Objects

To clear the hit count statistics for the named cache, use the command:

```
dx% clear cache <name> stats
```

This does not affect the object counts.

For example:

```
dx% clear cache secureImages stats
```

To clear all objects and statistics from the named cache, use the command:

```
dx% clear cache <name>
```

For example:

```
dx% clear cache secureImages
```

Delete Cache Commands

To delete a named cache, use the command:

```
dx% delete cache <name>
```

For example:

```
dx% delete cache secureImages
deleted
(*) dx%
```

NOTE: A cache cannot be deleted while it is still associated with a cluster. You must disassociate the cache from the cluster before deleting the cache.

Associate or Disassociate a Cluster with Cache

To associate a cluster with a named cache, use the command:

```
dx% set cluster <name> cache <name>
```

For example:

```
dx% set cluster fred cache secureImages
(*) dx%
```

The cache is disabled by default. To enable or disable caching for a cluster, use the command:

```
dx% set cluster <name> cache <name> [enabled | disabled*]
```

Clear Commands

To clear the association of a cluster with a cache, use the command:

```
dx% clear cluster <name> cache <name>
```

For example:

```
dx% clear cluster fred cache secureImages
(*) dx%
```

Show Cache Commands

To show the configuration for a cache, use the command:

```
dx% show cache [<name>]
```

If no name is specified, all caches are displayed. For example:

```
dx% show cache
Cache [m1]
Max Objects: 8192 (8.19K)
Size: 2097152 (2.00MB)
Used by cluster: m1
```

To display existing target server-like statistics, use the command:

```
dx% show cache <name> stats [<number> | LRU <number> | MRU <number> | content_type
detail | hit_count <number> | object_size | summary ]
```

This command shows detailed statistics on the object based on criteria selected. If no criteria is selected, the statistics for all criteria are shown. LRU is the "Least Recently Used" element, and MRU is the "Most Recently Used" element. Where the commands take an optional `<number> argument`, `<number>` limits the count of printed records. The valid range for `<number>` is 1-100, and the default is 10.

NOTE: The `show cache <name> stats` command can display the statistics for a maximum of 100 objects.

Some examples are:

```
dx% show cache secureImages stats object_size
Object Size Statistics:
  Object Size
    (bytes)      # Objects    # Hits
---------------- ------------ ------------
     1 - 256            0          0
   256 - 512            1         12
   512 - 1K             4         48
    1K - 2K             6         72
    2K - 4K             1         12
    4K - 8K             3         36
    8K - 16K            1         12
   16K - 33K            0          0
   33K - 66K            1         12
   66K - 131K           0          0
  131K - 262K           0          0
  262K - 524K           0          0
      1M+               0          0
```

```
dx% show cache secureImages stats content_type
Content-Type Statistics:
Content-Type # Objects # Hits
-------------------------------- ---------- ----------
image/jpeg 3 36
text/html 1 12
image/gif 13 156
```

```
dx% show cache secureImages stats hit_count 5
Size            # Hits          Cache Time      Order             URL
----------      ----------      ----------      ----------        ---------------------------------
    2K              12              321             1             /images/FossilLogo.gif
    3K              12              321             2             /images/bb120x30.jpg
    2K              12              321             3             /images/main_pg.gif
    1K              12              321             4             /images/yahoo_120X30.gif
    1K              12              321             5             /images/yahoo_10_61.gif
```

```
dx% show cache secureImages stats MRU 3
Size            # Hits          Cache Time      Order             URL
----------      ----------      ----------      ----------        ---------------------------------
    2K              12              321             1             /images/FossilLogo.gif
    3K              12              321             2             /images/bb120x30.jpg
    2K              12              321             3             /images/main_pg.gif
```

```
dx% show cache secureImages stats LRU 3
Size          # Hits        Cache Time    Order             URL
----------    ----------    ----------    ----------        ---------------------------------
    35K           12            323           1             /
    2K            12            323           2             /images/sh41.gif
    357           12            322           3             /images/sm.gif
```

NOTE: An expired object is not removed from the cache until it is explicitly requested (a “miss”), another object needs to get cached (causing the DX appliance to scavenge for space), or the operator removes it using the `clear cache` command. This means that the `show cache stats` command will occasionally include some expired, but not yet removed objects.

Show Cluster Cache Commands

To show Cluster Cache statistics, use the command:

```
dx% show cluster <name> cache stats [http | io ]
```

This command shows target host-like statistics relating to the traffic a cluster is routing to a cache. If `http` is specified, only the HTTP stats are shown. If “io” is specified, only the I/O stats are shown. If neither is specified, both sets are shown.

For example:

```
dx% show cluster m1 cache stats
IO Statistics - cluster m1 cache m1
Bytes In  (Resp from Cache)0B
Bytes Out (Inserts to Cache)0B

HTTP Statistics - m1
Responses from Servers:
**  Total 1XX Response Codes  **0
Response Code 1000
```

AppRules

AppRules are provided to enable or disable in-memory caching. To enable caching of objects using an AppRule, the syntax is:

```
cache “<seconds>”
```

For example:

```
PTH: http_reply_code eq “200” and url ends_with “.gif” then cache “<seconds>”
```

Usage

This section describes how the cache feature can be implemented and configured for normal usage, and how you can test the feature to ensure that it is working correctly.

Case 1

1. Add a cache using the command:

   dx% **add cache <name>**

2. Set the cache parameters using the command:

   dx% **set cache <name> max_object_size <bytes> ... commands**

3. Create an AppRule that will cache objects when they match the AppRule for caching. An example of an AppRule that caches PDF files is:

   `PTH: url ends with ".pdf" then cache "30" (time in seconds)`

To test the configuration, make an HTTP request to retrieve the HTTP object under test. Observe that the first request is retrieved from the origin server. For example, make a request to retrieve `JuniperCacheTestFile.pdf` using your Web browser. The first request for `JuniperCacheTestFile.pdf` must be retrieved from an origin server with or without caching.

Clear your Web browser cache and make another request for the same file. Make sure that the DX appliance does not retrieve this file from the origin server. One way to assure that outcome is to delete the file from the origin server.

If the request succeeds, the caching functionality is working. If you receive error 404 ("requested object does not exist on this server"), then the object was not cached.

Case 2

Specify an "expires" time for the cache. This is the time period after which the in-memory cache is invalidated. Run through the Normal Scenario (as described in "Case 1").

If you make the second request BEFORE the "expires" time, the cache should successfully return the object. If you receive error 404 ("requested object does not exist on this server"), then the caching functionality is not working, and the DX appliance is attempting to retrieve the object from the origin server.

Case 3

Specify an "expires" time for the cache. This is the time period after which the in-memory cache is invalidated. Run through the Normal Scenario (as described in "Case 1").

Make the second request AFTER the "expires" time. The cache should not have the object and since you have removed the file from the origin server, you should receive the HTTP error 404 ("requested object does not exist on this server"). If you

replace the object on the origin server, the request for the HTTP object should succeed.

NOTE: An exception can occur when an object has expired in the cache in the DX appliance but is still available on the server. If a client sends a request with an “If-Modified-Since” header, the DX appliance will bypass the cache (since the object is expired there) and go directly to the server to fetch the object. The server will return a code 304 (“Not Modified”), which the DX appliance will not cache. This means that all requests with an “If-Modified-Since” header will result in a server hit until the DX appliance receives a client request without an “If-Modified-Since” header.

Chapter 16

Application Rules Syntax

This chapter describes the DX Application Acceleration Platform with Application Rules Syntax, discussing the following topics:

Overview

This chapter provides an overview of the OverDrive Application Rules Translator (AppRules for short) feature. It describes how the OverDrive feature fits into the DX Application Acceleration Platform , along with all aspects of the feature (i.e., grammar, management, operation, etc.). OverDrive is an optional feature, and it requires a license key to work (see “The License Key” on page 47).

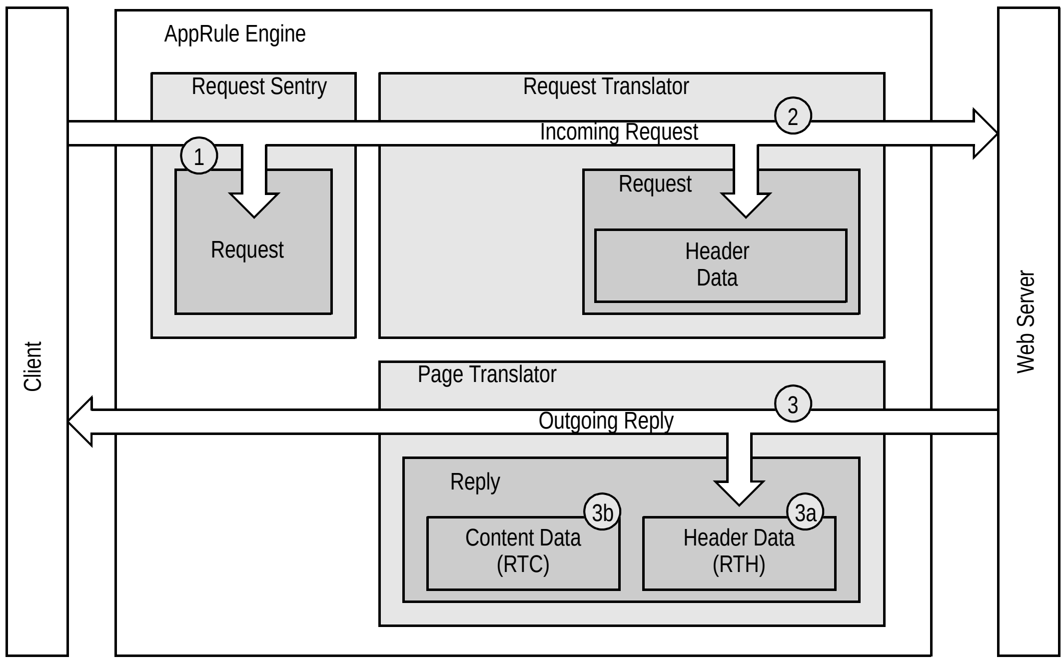
Basic Application Rule Concepts

Application Rules are simple rules written in plain language that are used to programmatically describe real-time changes that will be made to requests and replies passing through the DX appliance. This allows flexible applications that allow sites to respond to changing business needs. With AppRules, you can make automatic changes to user requests without making expensive changes to back-end applications. For example, you can route all requests for pictures (gif or jpg) to a particular server.

Application Rules also ensure request completion by re-initiating a request sequence based on parameters such as incoming client request headers, server

response information, or response content from the application. For example, you can automatically initiate a retry if there was an internal server error (HTTP error 500) or if the application returns a particular keyword such as "Unavailable" in its response. This is shown in Figure 53.

Figure 53: Application Rules General Categories



Application Rules are segmented into various types based on how and when they are processed within the DX appliance. When examined from a very high level, rules are either oriented around security-based connection management or around request and/or reply translations.

Application Rules can be applied to either incoming requests from the client, or to outgoing data or responses from the servers. Incoming requests are first processed through the "Request Sentry". The request sentry acts as the security gateway for all incoming requests, ensuring that they conform to specific criteria as defined in the Request Sentry (RS) AppRules (Step 1). Having been accepted, the request is then passed through to the Request Translator and the Request Translator Header AppRules (RTH). The request then passes to the target web server (Step 2).

Once the request has been processed by the origin server, the outgoing reply is passed through the Page Translator (Step 3). Like the Request Translator, the Page Translator has two components, the "Header Translation" (Page Translator Header, PTH, Step 3a) and the "Content Translation" (Page Translator Content, PTC, Step 3b).

Application Rule Anatomy

Application Rules themselves are roughly divided into two parts, "Test Conditions" and "Actions". A single rule can contain multiple test conditions and multiple actions (although some rules only allow a single action). Actions are executed only when all of the test conditions have been resolved as true.

Test conditions involve having the DX appliance compare various test operators against the data supplied in either the request or response with pre-determined values called "Test Variables". For example, you might declare test variables such as "URL" or "HTTP version." The DX appliance then tests the piece of data against this pre-defined value to evaluate its validity. For example, a test condition might be: "Is the URL for this request equal to /index.html?" If this condition is valid, then the test condition resolves as true.

Actions, on the other hand, involve changing something about the connection, the request, or the reply based upon the test condition. When test conditions and actions are placed together, you have a "rule". For example, a rule might be: "If the URL equals /index.html, then redirect the request to server http://www.myserver.com using the same URL as the one supplied in the request as the redirect URL".

Obviously, the actual rule syntax is less verbose than indicated here, but the same idea is conveyed. You must define what pieces of data can be analyzed (test variables), how they can be tested (test conditions), and what to do when the tests are true (actions).

Application Rule Execution

To keep consistent with AppRule categories, the DX appliance groups all AppRules together by category during execution. This way, when the "Request Sentry" rules execute, only those rules designated as being request sentry rules are evaluated.

Application Rules are executed in order. If request sentry #1 hits, the request sentry section of the rule set is done. If not, the second request sentry is tested. Whether a rule hits or the end of the Request sentry section comes, processing moves on to the "Request Translator Header" (RTH) rules. The same happens for "Page Translator Header" (PTH) rules passing to "Page Translator Content" (PTC) processing.

There are a few of exceptions to these rules are:

- There is a "continue" function built into RTH and PTH rules. If an RTH or PTH rule hits and ends with the "`and continue`" command, the next RTH or PTH rule in the list will be tested.
- PTC rules in a rule set will be tested and processed in order, no "continue" is needed.
- Only one Request Sentry (RS) rule can fire.

Experience has demonstrated that the arrangement shown in Table 16 makes the most sense.

Table 16: Application Rule Operation

Application Rule Type	Application Rule Execution Mode
Request Sentry (RS)	Exclusive
Request Translator Header (RTH)	Exclusive (Default)
Request Translator Content (RTC)	Collective
Page Translator Header (PTH)	Exclusive (Default)
Page Translator Content (PTC)	Collective

All rules execute in exclusive mode unless they are operating on content, in which case they run in the collective mode. Of course, because there are exceptions for every rule, the AppRule grammar provides mechanisms that allow rules running in Exclusive mode to run “semi” exclusively, and allow rules running in Collective mode to run “semi” collectively.

Application Rule Relationships

This section describes the various categories of application rules and the relationships between the test variables, test operators, and actions.

Request Sentry Application Rules

The request sentry rules operate at the connection level by allowing, denying, or possibly redirecting a request based on certain criteria. The various test variables, test conditions, and actions for the Request Sentry rules are described as follows.

Test Variables

The various entities within a client's request that must be available for analysis by request sentry rules are:

- URL
- Query String
- Request Header
- Request Cookie (individual)
- Any Header (headers as a unit)
- Any Request Cookie (cookies as a unit)
- HTTP Client Version
- HTTP Request Method
- Client IP Address
- Source IP Address
- SSL Cipherbits [1]
- SSL Ciphersuite

- SSL Version

Test Operators

The comparisons that can be performed against the pieces of data in a request are:

- Length, less than or greater than
- Less than or greater than [1]
- Length equals or does not equal
- Exists or does not exist
- Equals or does not equal (case sensitive/insensitive)
- Contains or does not contain (case sensitive/case-insensitive)
- Starts with or does not starts with (case sensitive/case-insensitive)
- Ends with or does not end with (case sensitive/case-insensitive)

Actions

The actions that can be taken by a request sentry rule when its test conditions all pass are:

- Close the connection by sending an RST or a FIN; interoperable only with the logging action.
- Redirect a request using an HTTP 302.
- Reply with an HTTP 404.
- Log that a rule passes and that its actions are being executed. The log function can be used at the end of any rule.

1.The "less than and greater than" test operators only work with the ssl_cipher_bits test variable, and the ssl_cipher_bits test variable only works with request sentry rules.

The relationships that are allowed between the request sentry test variables and the request sentry test operations are shown in Table 17.

Table 17: Request Sentry Test Variable and or Operator Matrix

Test Variable	Test Operator							
	(not) exists	(not) equals	(not) contains	(not) ends with	(not) starts with	length less than/greater than	less than/greater than	length equals/not equals
url		X	X	X	X	X		X
query string	X	X	X	X	X	X		X
(any) request header	X	X	X	X	X	X		X
HTTP Version		X						
HTTP Method		X						
client IP Address		X	X	X	X			
source IP Address		X					X	
ssl_cipher_bits							X	
ssl_ciphersuite		X	X	X	X			
ssl_version		X	X	X	X			

There is no qualification regarding the actions and their relationship to the test conditions. Any legal request sentry test condition can be used in combination with any legal request sentry action. As for the relationship of actions between one another in a single rule, the relationships shown in Table 18 exist.

Table 18: Request Sentry Action Matrix

Action	Action			
	close connection	redirect (reply 302)	reply with 404	log
close connection	X			X
redirect (reply 302)		X		X
reply with 404			X	X
log	X	X	X	X

As you can see, each of the connection-handling actions must stand alone, but may be used in conjunction with the logging action.

Request Translator Application Rules

Request Translator AppRules are designed to modify incoming requests at either the header level or the content level. The header and content Request Translator rules are shown separately.

Request Translator Header Application Rules

The Request Translator Header (RTH) rules operate on the HTTP header segment of the incoming request. This includes the URL and query string, along with the headers that may be part of the request. The test variables, test conditions, and actions for RTH rules are described as follows.

Test Variables

The entities within a client's request that must be available for analysis by an RTH rule are:

- URL
- Query String
- Request Header
- Request Cookie (individual)
- Http Client Version
- Http Request Method
- Client Ip Address
- Source Ip Address

Test Operators

The comparisons that can be performed against the pieces of data are:

- Exists or does not exist
- Equals or does not equal (case sensitive/insensitive)
- Contains or does not contain (case sensitive/case-insensitive)
- Starts With or does not start with (case sensitive/case-insensitive)
- Ends With or does not end with (case sensitive/case-insensitive)

Actions

The actions that can be taken for a RTH rule when its test conditions all pass are:

- Insert a new request header
- Insert a new request cookie
- Update an existing request header

- Update an existing request cookie
- Delete an existing request header
    - Delete an existing request cookie
    - Append data to the test variable value either just after the location where the search string match was made or at the end of the variable data.
    - Replace existing data in a test variable with new data or replace just the matched search string within the variable.
    - Prepend data to a test variable value either just before the location where the search string match was made or at the beginning of the variable data.
    - Redirect an incoming request using either the client's URL or a pre-determined URL allowing for HTTP or HTTPS, along with an HTTP port and possible prepended URL path information.

The relationships between the RTH test variables and the RTH test operations that are allowed are shown in Table 19.

Notice that all of the actions can interoperate with all of the test variables with the exception of append, replace, and prepend actions which cannot interoperate with the HTTP version, HTTP method, and/or the client IP address. Note also that a cookie header cannot be operated on directly; a rule operates on the individual cookies. The relationship between the actions is such that all actions can interoperate with one another, however, if a rule contains the redirect action, that action must be last in the list of actions.

Table 19: Request Translator Header Test Variable and Operator Matrix

Test Variable	Test Operator					
url		X	X	X	X	
query string	X	X	X	X	X	
request header	X	X	X	X	X	
request cookie	X	X	X	X	X	
HTTP Version		X				
HTTP Method		X				
client IP Address		X	X	X	X	
source IP address		X				X
	(not exists)	(not) equals	(not) contains	(not) ends with	(not) starts with	less than/greater than

The actions and their relationship to the test variables that are allowed are shown in Table 20.

Table 20: Request Translator Header Action and Test Variable Matrix

Action	Test Variable						
	url	query string	request header	request cookie	http version	http method	client IP address
insert header	X	X	X	X	X	X	X
insert cookie	X	X	X	X	X	X	X
update header	X	X	X	X	X	X	X
update cookie	X	X	X	X	X	X	X
delete header	X	X	X	X	X	X	X
delete cookie	X	X	X	X	X	X	X
append	X	X	X	X			
replace	X	X	X	X			
prepend	X	X	X	X			
redirect	X	X	X	X	X	X	X
route_request	X	X	X	X	X	X	X

Request Translator Content Application Rules

The main difference between the RTH rules and the RTC rules is the introduction of the content test variable in the content rules. Special consideration must be made when dealing with this variable relative to the other test variables, especially with regard to actions. The content variable is the only variable that can be used with an RTC action. All of the other test variables are used in test conditions only. The various variables, operators, and actions required for the RTC rules are described as follows.

Test Variables

The following are the various entities within a client's request that must be available for analysis by an RTC rule:

- URL
- Query String
- Request Header
- Request Cookie (individual)
- Http Client Version

- Http Request Method
- Client Ip Address
- Content

Test Operators

The various comparisons that can be performed against the pieces of data in a request and pre-defined values are:

- Exists or does not exist
- Equals or does not equal (case sensitive/insensitive)
- Contains or does not contain (case sensitive/insensitive)
- Starts with or does not start with (case sensitive/insensitive)
- Ends with or does not end with (case sensitive/insensitive)

Actions

The actions that can be taken for an RTC rule when its test conditions all pass are:

- Append data to the content just after the location where the search string match was made.
- Replace existing data in the content with new data at the location where a specific search string matched.
- Prepend data to the content just before the location where the search string match occurred.

NOTE: For RTC rules, the only variable that may be operated upon in the actions is the “content variable”. All other variables may be used for reference in determining whether a rule is true (i.e., they can be used in test conditions), but not in the actions.

The allowed relationships between the RTC test variables and the RTC test operations are similar to header rules and are shown in Table 21.

Table 21: Request Translator Content Test Variable and Operator Matrix

Test Variable	Test Operator: (not exists)	(not) equals	(not) contains	(not) ends with	(not) starts with
url		X	X	X	X
query string	X	X	X	X	X
request header	X	X	X	X	X
request cookie	X	X	X	X	X
HTTP Version		X			
HTTP Method		X			
client IP Address		X	X	X	X
content			X		

The actions and their relationship to the test variables that are allowed are shown in Table 22.

Table 22: Request Translator Content Action and Test Variable Matrix

Action	Test Variable: url	query string	request header	request cookie	request cookie header	http version	http method	client IP address	content
append									X
replace									X
prepend									X

NOTE: Actions only operate on the content variable; all other variables are used for reference within test conditions. The relationship between the actions is such that only one action can be performed per content rule (e.g., prepend, append, or replace). This is fundamentally different from the request sentry or the header-oriented rules where multiple actions can be executed in a single rule.

Page Translator Application Rules

Page Translator rules are designed to modify outgoing replies at either the header level or the content level. We will examine the header and content Page Translator rules separately.

Page Translator Header

The purpose of the Page Translator Header rules is to modify the outgoing HTTP reply headers based on certain test conditions. The variables, operators, and actions required for this type of rule are described as follows.

Test Variables

The entities within a server's response that must be available for analysis by a Page Translator Header rule are:

- URL
- Query String
- Request Header
- Request Cookie (individual)
- Reply Header
- Reply Cookie (Individual)
- Http Reply Code
- Http Client Version
- Http Request Method
- Client Ip Address
- Source Ip Address

NOTE: Unlike the Request Translator Header, Page Translator Header rules can use both the request information and the reply information in test conditions.

Test Operators

The comparisons that can be performed against the pieces of data in either the request or the reply are:

- Exists or does not exist
- Equals or does not equal (case sensitive/insensitive)
- Contains or does not contain (case sensitive/insensitive)
- Starts with or does not start with (case sensitive/insensitive)
- Ends with or does not end with (case sensitive/insensitive)

Actions

The actions that can be taken for a PTH rule when its test conditions all pass are:

- Insert a new reply header
- Insert a new reply cookie
- Update an existing reply header
- Update an existing reply cookie
- Delete an existing reply header
- Delete an existing reply cookie
    - Append data to a reply test variable value either just after the location where the search string match was made or at the end of the variable itself.
    - Replace existing data in a reply test variable with new data or replace just the search string within the variable.
    - Prepend data to a reply test variable value either just before the location where the search string match was made or at the beginning of the variable itself.
    - Cache the requested file in In-Memory Cache within the DX appliance (refer to "3G Cache" on page 235).
    - Continue to run another rule even though the actions for the current rule have been run (except for I/O-based actions such as request retries).

The relationships allowed between the RTH test variables and the RTH test operations are shown in Table 23.

Table 23: Page Translator Header Test Variable and Operator Matrix

Test Variable	Test Operator					
	(not) exists	(not) equals	(not) contains	(not) ends with	(not) starts with	less than/greater than
url		X	X	X	X	
query string	X	X	X	X	X	
request header	X	X	X	X	X	
request cookie	X	X	X	X	X	
reply header	X	X	X	X	X	
reply cookie	X	X	X	X	X	
HTTP reply code		X			X	
HTTP Version		X				
HTTP Method		X				
client IP Address		X	X	X	X	
source IP Address		X	X	X	X	X

The actions and their relationship to the test variables that are allowed are shown in Table 24.

Table 24: Page Translator Header Action and Test Variable Matrix

Action	Test Variable								
cache	X								
insert reply header				X	X				
insert reply cookie				X	X				
update reply header				X	X				
delete reply header				X	X				
delete reply cookie				X	X				
append				X	X				
replace				X	X				
prepend				X	X				
retry_request	X	X	X	X	X	X	X	X	X
	url	query string	request header	reply header	reply cookie	http reply code	http version	http method	client IP address

Notice that all of the actions are based upon a generic reply header. The other test variables are solely available for test conditions; they cannot be altered by Page Translator Header rules. The relationship between the actions is such that all actions can interoperate with one another.

Page Translator Content

Page Translator Header (PTH) rules use all of the same test variables as Page Translator Content (PTC) rules, and also have the *content* test variable. Special consideration must be made when dealing with this variable relative to the other test variables, especially with regard to actions. The actions in a Page Translator Content rule operate only on the content variable and no other. The other variables are merely used for test conditions to determine if the content should be changed in some way.

WARNING: Care must also be taken when writing these rules. They will change **any** matching string in the code.

Test Variables

The entities within a client's request that must be available for analysis by a PTC rule are:

- Url
- Query String
- Request Header
- Request Cookie (individual)
- Reply Header
- Reply Cookie (Individual)
- Http Reply Code
- Http Client Version
- Http Request Method
- Client Ip Address
- Content

Test Operators

The comparisons that can be performed against the pieces of data in a request and pre-defined values are:

- Exists or does not exist
- Equals or does not equal (case sensitive/insensitive)
- Contains or does not contain (case sensitive/insensitive)
- Starts with or does not start with (case sensitive/insensitive)
- Ends with or does not end with (case sensitive/insensitive)

Actions

The actions that can be taken for a PTC rule when its test conditions all pass are:

- Append data to the content just after the location where the comparison was made.
- Replace existing data in the content with new data at the location where a specific search string matched.
- Prepend data to the content just before the location where the comparison was made.

NOTE: For Page Translator Content rules, the only variable that may be operated upon in the actions is the content variable. All other variables may be used for reference in determining whether a rule is true (i.e., they can be used in test conditions), but not in actions.

The allowed relationships between the PTC test variables and the PTC test operations are similar to the header rules and are shown in Table 25.

Table 25: Page Translator Content Test Variable and Operator Matrix

Test Variable	Test Operator				
url		X	X	X	X	
query string	X	X	X	X	X	
request header	X	X	X	X	X	
request cookie	X	X	X	X	X	
reply header	X	X	X	X	X	
reply cookie	X	X	X	X	X	
HTTP reply code		X			X	
HTTP Version		X				
HTTP Method		X				
client IP Address		X	X	X	X	
source IP Address		X	X	X	X	X
content			X			
	(not) exists	(not) equals	(not) contains	(not) ends with	(not) starts with	less than/greater than

The actions and their relationship to the test variables that are allowed are shown in Table 26.

Table 26: Page Translator Content Action and Test Variable Matrix

Action	url	query string	request header	request cookie	reply header	reply cookie	http version	http method	client IP address	content
append										X
replace										X
prepend										X
retry_request [1]	X	X	X	X	X	X	X	X	X	X

1.For the retry_request action to work correctly with Page Translation Contents, the factory setting `fcl` must be explicitly enabled (it is disabled by default). Contact your Juniper Administrator.

NOTE: Notice that actions only operate on the content variable; all other variables are used only for reference within test conditions. The relationship between the actions is such that only one action can be performed per content rule (e.g., prepend, append, or replace). This differs from the header-oriented rules and the request sentry where multiple actions may be specified per rule.

Application Rule Grammar

This section describes the grammar for Application Rules. It supports the various rules described in the previous sections, and also describes how each keyword operates at runtime.

Application Rule Syntax

The basic syntax for AppRules can be described as follows:

```
<rule_type>: <test_condition> [and <test_condition>...] then <action> [and <action>     ]
```

where:

- `<rule_type>` designates a mnemonic (typically a two or three letter abbreviation) correlating to a specific rule class (which is then terminated with a colon).

- `<test_condition>` specifies a particular test condition statement. There may be more than one, with each one separated by the keyword "`and`".

- `<action>` designates the action that is performed when all test conditions for a certain rule have been met. Although some AppRules only allow one action, those that allow more than one should have each action statement separated by the keyword "`and`".

It is customary to separate each logical component by some amount of arbitrary whitespace, although this is not required. Single line comments can be placed in the ruleset by placing a "`#`" at the beginning of the line.

For example:

```
# This is a comment.
```

All arguments must be enclosed in double quotes and cannot span across lines. Single quotes cannot be used. Escape characters can be used in limited cases.

Application Rule Types

Currently, there are three supported rule types, with one of the rule types having sub-types:

- Request Sentry (RS)
- Request Translator Header (RTH)
- Page Translator
  - Page Translator Header (PTH)
  - Page Translator Content (PTC)

Test Conditions

Each test condition is formatted using the following syntax:

```
<variable_statement> <operator> [sub_operator] [argument]
```

where:

- `<variable_statement>` is either the name of the variable itself, or the variable type and then a variable name.
- `<operator>` indicates the type of test operator to use against the variable in conjunction with the argument.
- `[sub_operator]` is an optional value that may be used with certain operators to further qualify how the operator is used.
- `[argument]` is the value to test against the current variable value. Not all operators require an argument.

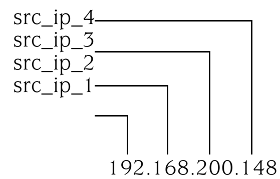
Variables

The variables/variable types shown in Table 27 are supported.

Table 27: Variables or Variable Types that are Supported

Variable	Description
url	While not technically accurate in its name, this indicates the URL of the HTTP request.
query_string	This is the portion of the URL that exists after the '?'. If the '?' is not present in the URL, then the <query_string> is considered not to exist.
request_header "<header_name>"	Specifies a variable type as request_header with the specific variable name being <header_name> as enclosed in required double quotes. A header name can contain only these characters, A-Z, a-z, 0-9, '-', '_', and should not refer to a "Cookie" header since they are treated separately.
any_request_header	Refers to any of the HTTP headers in the request (not including Cookie headers since they are treated separately).
request_cookie "<cookie_name>"	Specifies a variable of type <request_cookie> with the specific variable name being <cookier_name> as enclosed in the required double quotes. A cookie name can contain only the following characters: A-Z, a-z, 0-9, '-', '_'. The request_cookie examines the "Cookie" HTTP headers for the specified cookie name.
any_request_cookie	Refers to any individual cookie name/value pair in the request.
http_request_version	Specifies the HTTP version of the request (1.0 or 1.1).
reply_header "<header_name>"	Specifies a variable type as <reply_header> with the specific variable name being <header_name> as enclosed in required double quotes. A header name can contain only these characters, A-Z, a-z, 0-9, '-', '_', and should not refer to a "Set-Cookie" header since they are treated separately.
reply_cookie "<cookie_name>"	Specifies a variable of type <reply_cookie> with the specific variable name being <cookie_name> as enclosed in the required double quotes. A cookie name can contain only the following characters: A-Z, a-z, 0-9, '-', '_'. The reply_cookie examines the "Set-Cookie" HTTP headers for the cookie name.
http_reply_code	Refers the HTTP code (200, 404, 502, etc.) that appears in the reply.
http_reply_version	Specifies the HTTP version of the reply (1.0 or 1.1).
http_method	This indicates the HTTP method used in the request (i.e., Post, Get, Head, etc.).
src_ip, sip	This is the client's IP address and is also known as the source IP address. All comparisons are made against IPv4 dot notation addresses, so test arguments should be made accordingly.

Table 27: Variables or Variable Types that are Supported

Variable	Description
src_ip_1, src_ip_2, src_ip_3, src_ip_4	These test variables correspond to the four octets composing an IPv4 network address. The numeric designator indicates the octet that is being referenced. The first octet is considered the "class A" octet, second octet is the "class B" one, and so forth. For example:  This allows for fine-grained checking of source IP addresses on a per-octet level. You can specify ranges of IP addresses for which a rule applies. This makes it easier to "classify" many users much like subnetting does.
content	This refers to any ASCII-based data located in an HTTP response after the HTTP headers. This variable will only operate with the contains and ci_contains test condition operators (see below).
ssl_cipher_bits	This variable refers to the size of the key. It can be up to four digits (less than 1024).
ssl_cipher_suite	This variable indicates the list (suite) of ciphers.
ssl_version	The variable shows the version of Secure Socket Layer (SSL) that is supported. The value is case-sensitive, and must be entered in the form: SSLv2, SSLv3 or TLSv1 instead of ssvl3.

In the variables shown in Table 27, only the request or reply cookie and header variables use the variable type or variable name style; the others are all specific variables.

Table 28 shows a list of the valid <header_name> values. These values are all case-sensitive.

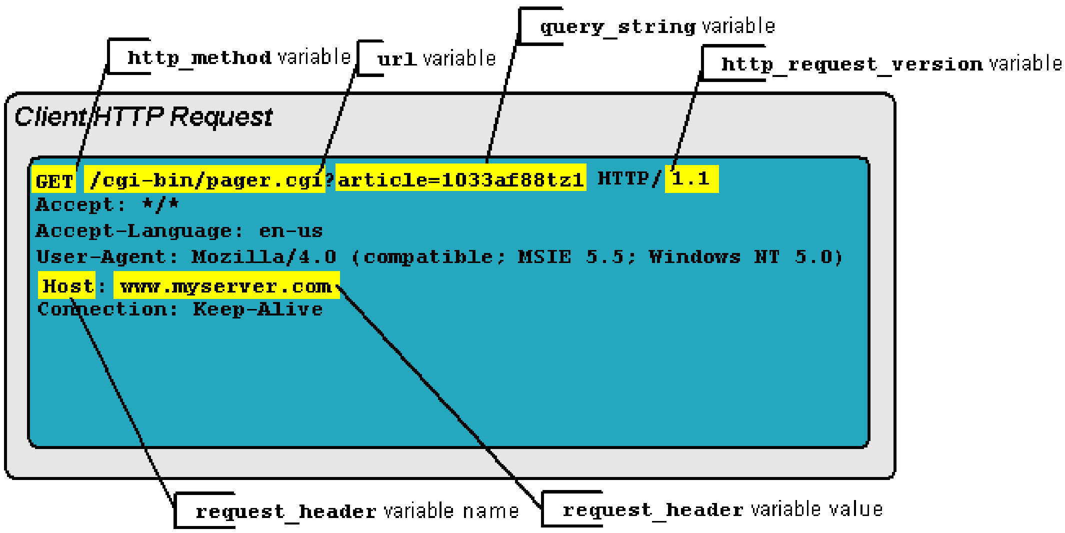
Table 28: Valid Header Variables

Header Variable	Header Variable	Header Variable
Date	Apply-To-Redirect-Ref	Proxy-Connection
Server	Max-Forwards	Content-Length
Accept-Ranges	Proxy-Authorization	Content-Type
Last-Modified	Age	Content-Encoding
ETag	Proxy-Authenticate	Transfer-Encoding
Cache-Control	Public	Via
Pragma	Retry-After	Warning
Expires	Content-Base	Vary
Connection	Content-Language	Accept-Encoding
Keep-Alive	Content-Location	User-Agent
Range	Content-MD5	Host
Referer	Content-Range	If-None-Match
If-Modified-Since	DASL	Location
Cookie	Redirect-Ref	Authorization
Cookie2	Brief	WWW-Authenticate
Set-Cookie	Call-Back	Trailer
Set-Cookie2	Notification-Delay	TE
FRONT-END-HTTPS	Notification-Type	Expect
From	Range	DAV
Allow	Subscription-ID	Depth
Upgrade	Subscription-Lifetime	Destination
Accept	Transaction	If
Accept-Charset	Label	Lock-Token
Accept-Language	Ordered	Overwrite
If-Match	Position	Status-URI
If-Range	Allow-Rename	Timeout
If-Unmodified-Since	X-Powered-By	

NOTE: All arguments must be enclosed in double quotes. Be sure to use a pure text editor such as Wordpad when creating the arguments. The quotes used by some word processors are not processed by the AppRule engine. A single argument may not spand more than one line. Certain “escape” characters are allowed in limited cases.

Figure 54 shows the relationship between an incoming client's HTTP request and the AppRule variables specified in a given rule.

Figure 54: Client HTTP Request and Application Rules Variable Relationship

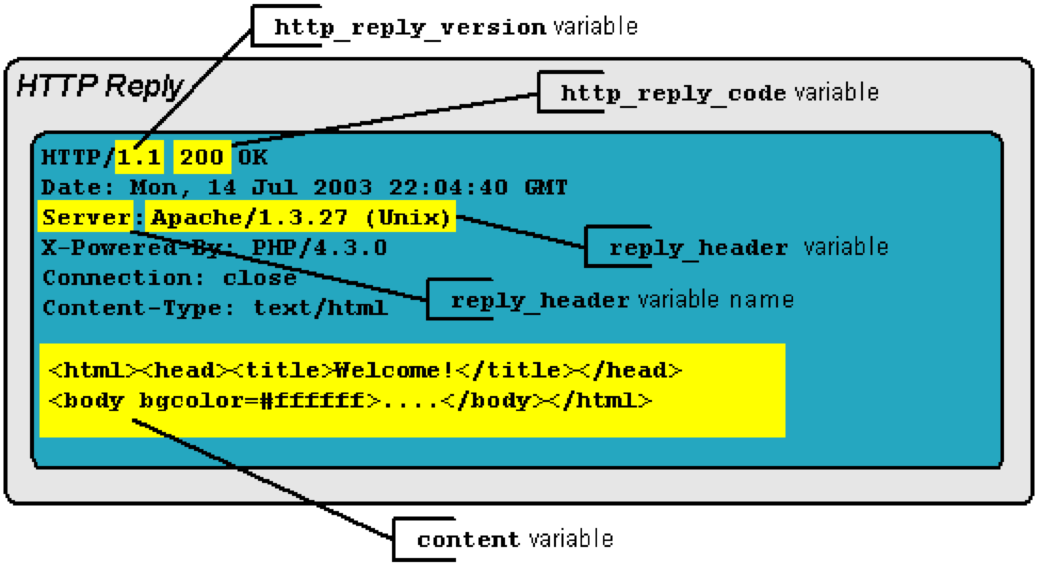


Note that in the case of the header variable name and value, you can specify any "well-known" header in the request. For example, you might have a language translation rule like this:

```
RTH: header "Accept-Language" equals "fr-ca" then update_header "Host" "french-canada.myserver.com"
```

This rule causes the host to change, based on the language in the HTTP request. In the example shown in Figure 54, you would send all browsers that are indicating a French Canadian user (language code fr-ca) to the virtual host `french-canada.myserver.com`. The content variable includes any data returned back to the user after the headers, as shown in Figure 55.

Figure 55: HTTP Reply and Application Rules Variable Relationship



The content variable is used only by the content-oriented AppRules (Request Translator Content and Page Translator Content).

Operators

The operators shown in Table 29 are used when formulating test conditions.

Table 29: Operators Used When Formulating Test Conditions

Operator	Description
ex, exists	Is true if the variable's value exists; no test argument is necessary. Applicable only to header variables.
nx, not_exists	True if the variable's value is not present in the request; no test argument is necessary. Applicable only to header variables[1].
eq, equals	True if there is an exact (case-sensitive) match between a variable's value and the test argument.
ci_eq, ci_equals	True if there is a case-insensitive exact match between a variable's value and the test argument.
ne, not_equals	True if the variable's value is not equal to the test argument.
ci_ne, ci_not_equals	True if the variable's value is not equal to the case-insensitive test argument.
c, contains	True if variable's value contains the argument within it.
ci_c, ci_contains	True if the variable's value contains the case-insensitive argument within it.
nc, not_contains	True if the variable's value does not contain the test argument.
ci_nc, ci_not_contains	True if the variable's value does not contain the case-insensitive test argument.
sw, starts_with	True if the variable's value starts with the specified argument.

Table 29: Operators Used When Formulating Test Conditions

Operator	Description
ci_sw,ci_starts_with	True if the variable's value starts with the specified case-insensitive argument.
ns, not_starts_with	True if the variables value does not start with the test argument.
ci_ns, ci_not_starts_with	True if the variable's value does not start with the case-insensitive test argument.
ew, ends_with	True if the variable's value ends with the specified argument.
ci_ew, ci_ends_with	True if the variable's value ends with the specified case-insensitive argument.
nw, not_ends_with	True if the variable's value does not end with the test argument.
ci_nw, ci_not_ends_with	True if the variable's value does not end with the case-insensitive test argument.
l_gt, length_greater_than	True if the variable's value is not longer than the test argument numeric value (as specified in bytes)
l_lt, length_less_than	True if the variable's value is less than the test argument numeric value (as specified in bytes).
l_eq, length_equals	True if the variable's value is exactly the same length as the test argument numeric value (as specified in bytes).
l_ne, length_not_equals	True if the variable's value is not the same length as the test argument numeric value (as specified in bytes).
greater_than	True if the variable's value is greater than the test argument's numeric value.
less_than	True if the variable's value is less than the test argument's numeric value.

1.The semantic of "any_request_header" does not mean "any one request header" when used with the "not_exists" operator. This application rule is considered to be "all request headers as a group." The implication is that the test condition: "any_request_header not_exists" will treat all the request headers as a group and determine their non-existence as a group.

As an example, "GET / HTTP/1.0\r\n \r\n" will get result in the test condition returning TRUE, while the request "GET / HTTP/1.0\r\n Host: www.xyz.com\r\n \r\n" will result in the test condition returning FALSE, because as a group the request headers do exist.

Note that each operator has a long syntax and a short syntax; both are equally valid and can be used interchangeably.

Arguments

All test arguments must be enclosed within double quotes and cannot span more than a single line. Character restrictions are placed on the test argument depending upon the test variable as shown in Table 30.

Table 30: Arguments

<table>
<tr><th>Test Variable</th><th>Acceptable Characters</th></tr>
<tr><td>url test arguments</td><td>A-Z, a-z, 0-9, '%', '.', '/', '_', '-',',','~'</td></tr>
<tr><td>http_version test arguments</td><td>0.9, 1.0, 1.1</td></tr>
<tr><td>request/reply_header variable test arguments</td><td>A-Z, a-z, 0-9, space, '(', ')', '/', '.', ',', '_', '-',
';','~','%',':','>','+','<'</td></tr>
<tr><td>request/reply_header names</td><td>A-Z, a-z, 0-9, '_', '-'</td></tr>
<tr><td>cookie variable test arguments</td><td># A-Z, a-z, 0-9, space, '(', ')', '/', '\', '.', ',', '_', '-',
';','~','%',':','>','+','<','!', '@', '#', '$','&', '`',
apostrophe('),'?','[', ']', '^', '*','{', '}', '|','='</td></tr>
<tr><td>cookie names</td><td># A-Z, a-z, 0-9, '(', ')', '/', '\', '.', ',', '_', '-',
';','~','%',':','>','+','<', '!', '@', '#', '$', '&', '`',
apostrophe('), '?','[', ']', '^', '*', '{', '}', '|'</td></tr>
<tr><td>content test arguments</td><td>A-Z, a-z, 0-9, space, ',', ':', '`', '!', '@', '#', '$', '%', '^', '&', '(', ')', '=', '+', '[', ']', '{', '}', '|', ';', '"', '<', '>', '?', '~', '\' (as escape
character), '.', '/', '_', '-',

escaped characters: newline (\n), carriage return (\r), double quote (\"), asterisk (*), backslash (\\),

wildcard character '*'</td></tr>
<tr><td>length</td><td>The length test argument is limited to a range from 1 to 99999.</td></tr>
</table>

The content test argument offers some additional functionality not present in the other variables; namely, the ability to escape certain characters and the use of a wildcard character. The wildcard character “*” (star) represents zero to 26 arbitrary characters and can be present only within the content test argument (i.e., a content test argument cannot begin or end with an unescaped “*”). The argument as a whole is considered a search term.

Action Statements

Each of the action statements, because of their variety, has its own unique syntax. Table 31 is a list of all the actions with their syntax and descriptions of what that action does.

Table 31: Action Statements

Action	Description
insert_request_header “<header_name>” “<header_value>”	The insert_request_header action is used to insert a new, previously non-existing HTTP header as defined by <header_name> with the value <header_value> into the request. If the header already exists in the request, then the old one is first deleted and the new one inserted. This action is only available in the Request Translator Header rule type.
update_request_header “<header_name>” “<header_value_1>” [“<header_value_2>” . . ."<header_value_N>”]	The update_request_header action alters the value of the existing request header as designated by <header_name> with one or more values, <header_value_1> to <header_value_N>. If more than one value is specified, each value is used in a round-robin fashion. If the header <header_name> does not exist in the client request, then no action is performed (this is the primary difference between insert_request_header and update_request_header). This action is only available in the Request Translator Header rule type.
delete_request_header “<header_name>”	The delete_request_header action removes a request header from the client request matching <header_name>. This action is only available in the Request Translator Header rule type.
insert_request_cookie “<name>” “<value>”	This action injects an additional cookie with <name> and <value> into the client's request. This action is only available in the Request Translator Header rule type.
delete_request_cookie “<name>”	Removes the cookie corresponding to <name> if found in the client request. This action is only available in the Request Translator Header rule type.
insert_reply_header “<header_name>” “<header_value>”	The insert_reply_header action is used to insert a new, previously non-existing HTTP header as defined by <header_name> with the value <header_value> into the outgoing reply. If the header already exists in the reply, then the old one is first deleted and the new one inserted. This action is only available in the Page Translator Header rule type.
update_reply_header “<header_name>” “<header_value_1>” [“<header_value_2>” . . “<header_value_N>”]	The update_reply_header action alters the value of the existing reply header as designated by <header_name> with one or more values, <header_value_1> to <header_value_N>. If more than one value is specified, each value is used in a round-robin fashion. If the header <header_name> does not exist in the outgoing reply, then no action is performed (this is the primary difference between insert_reply_header and update_reply_header). This action is only available in the Page Translator Header rule type.

Table 31: Action Statements

Action	Description
delete_reply_header "<header_name>"	The delete_reply_header action removes a reply header from the outgoing reply matching <header_name>. This action is only available in the Page Translator Header rule type.
insert_reply_cookie "<name>" "<value>" "<domain>" "<path>" ["<expires_date>"] [secure]	This action injects an additional cookie with <name> and <value> into the client's request that will be valid for the given <domain> and <path>. Optionally, the expiration date and/or secure flag may be indicated. The <domain> value must contain either two or three dots in it (juniper.net should be .juniper.net). The path must begin with "/". Note that the <expires_date> must be Wdy, DD-Mon-YYYY HH:MM:SS GMT. The secure flag indicates that the cookie should only be sent over an SSL connection. This action is only available in the Page Translator Header rule type. The <expires_data> can also be an integer specifying the number of seconds from the time the cookie is inserted that it will expire.
append <variable> [term] "<append_value>"	The append action inserts <append_value> either just after the <variable>'s value or just after the search string match point within <variable>'s value. If the term keyword is used, then the append operation occurs at the search string match point; otherwise, it occurs at the end of the <variable>'s value. An additional requirement may exist where the <variable> must be a valid <variable> that exists in one of the test conditions within the rule where the append action appears and that same test condition must employ certain test conditions to be valid. See "Prepend, Append, Replace (PAR) Conditions" on page 276 for more information. When the append action is used in conjunction with the content variable, the term keyword must be present (appending data to the end of content data makes no sense since the content data is being streamed in indeterminately-sized packets). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.
prepend <variable> [term] "<prepend_value>"	The prepend action inserts <prepend_value> either at the start of <variable>'s value or just before the search string match point within the <variable>'s value. If the term keyword is used, then the prepend operation occurs at the point in the test condition <variable>'s value where it is evaluated as being true (i.e., where the search string matched). An additional requirement may exist where the <variable> must be a valid <variable> that exists in one of the test conditions within the rule where the prepend action appears and that same test condition must employ certain test conditions to be valid. See "Prepend, Append, Replace (PAR) Conditions" on page 276 for more information. When the prepend action is used in conjunction with the content variable, the term keyword must be present (it makes no sense to prepend content data at the beginning of the content as the content being sent back to the client is not entirely buffered--the data is streamed out in packets). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.

Table 31: Action Statements

<table>
<tr><th>Action</th><th>Description</th></tr>
<tr><td colspan="2">replace <variable> [term] "<replace_value>"</td></tr>
<tr><td></td><td>The replace action inserts <replace_value> either in place of the complete <variable>'s value or just the search string that matched within the <variable>'s value. If the term keyword is used, then the replace operation overwrites just the matched search string. An additional requirement may exist where the <variable> must be a valid <variable> that exists in one of the test conditions within the rule where the replace action appears and that same test condition must employ certain test conditions to be valid. See "Prepend, Append, Replace (PAR) Conditions" on page 276 for more information. When the replace action is used in conjunction with the content variable, the term keyword must be present (it makes no sense to replace content data since the content being sent back to the client is not entirely buffered--the data is streamed out in packets making for an indeterminate replacement based on the amount of data in a packet at any given moment). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.</td></tr>
<tr><td colspan="2">close_conn <RST|FIN></td></tr>
<tr><td></td><td>Closes the client connection and sends either a FIN packet or an RST, depending on which is specified.</td></tr>
<tr><td colspan="2">redirect "http[s]://host[:<port>][/<prepend_path>]" ["<URL>"]</td></tr>
<tr><td></td><td>Returns an HTTP 302 (redirect) reply to the client using the specified protocol/host and optional <port>, <prepend_path>, and <URL> (in actuality, it's the URI) as the Location header value. If the <URL> is not specified, then the URL from the client's request is used in its stead. If the optional <prepend_path> is specified, then whatever <URL> is used (either explicitly stated in the action or taken from the client's request) is prepended with that value.</td></tr>
<tr><td colspan="2">reply 302 "http[s]://host[:<port>][/<prepend_path>]" ["<URL>"]</td></tr>
<tr><td></td><td>Synonym for the redirect command.</td></tr>
<tr><td colspan="2">reply 404 [<404_file>]</td></tr>
<tr><td></td><td>Returns an HTTP 404 (not found) message using the content from the <404_file> as the body of the 404 message. The <404_file> must be imported onto the DX appliance using the Capture File command, and the argument string can have the following allowed characters: a-z, A-Z, 0-9, '-', '_', '.' For example:
RS: url sw "/" then reply 404 "my404.html"</td></tr>
<tr><td colspan="2">log</td></tr>
<tr><td></td><td>When specified, the rule that is executed is logged. This action is available only with the Request Sentry rule type.</td></tr>
<tr><td colspan="2">route_request target_host "<ip:port>"
route_request target_host "<ip1:port1>" ["<ip2:port2>"] ... ["<ipN:portN>"</td></tr>
<tr><td></td><td>The route_request AppRule for the Request Translator Header (RTH) allows users to route a request if an incoming request meets a test condition. You can specify the individual target host, a list of target hosts, or a group of target hosts. For more information, see "Route Request Application Rules" on page 289.</td></tr>
</table>

Table 31: Action Statements

Action	Description
retry_request [same | nosame | all] "number" and log	
	The retry_request AppRule allows users to retry a request if the response code for a previous request meets a test condition. For more information, see "Request Retry, Alerting, and Log (Transaction Assurance) AppRules" on page 289.
continue	
	This is a special action that does not alter the request in any way. Rather, it is used to override the default behavior for how RTH and PTH rules are executed. When this action is present, the subsequent rule in the rule set will be executed. This allows for a logical AND behavior to exist across individual rules in a rule set. Note that this action cannot be used with any rule that contains an I/O-based action (for example, redirect, retry_request, or route_request). A continue action may not exist by itself since it does not add any additional value to the rule (it would only act as an AND operation and the DX appliance already supports multiple test conditions with the and keyword).

NOTE: Multiple action statements can exist for a single AppRule in some instances. Each action is separated by the keyword "`and`".

Prepend, Append, Replace (PAR) Conditions

The Prepend, Append, and Replace (PAR) actions have unique inter-operations with test conditions, especially when the term keyword is used within the action. Whenever the term keyword is employed in a PAR operation, the variable that it is being updated must have a corresponding test condition with that same variable, and must use one of the test operators shown in Table 32, depending upon the rule type.

Table 32: PAR Test Operators

Rule Type	Variable	Valid Test Operators
Request Translator Header	url query_string request_header request_cookie	(ci_)contains (ci_)ends_with (ci_)starts_with
Page Translator Header	reply_header reply_cookie	(ci_)contains (ci_)ends_with (ci_)starts_with
Page Translator Content	content	(ci_)contains

The reason these relationships must exist is because the term keyword effectively means, “Do a PAR operation on the term that you matched against in the first test condition that references this variable with a valid test operator.” Therefore, you must restrict the test operators in the test condition to be those operators that yield a sub-string match.

If the term keyword is not used in a PAR action, then the test conditions within that rule do not need to reference the same variable used in the PAR action.

For example:

```
RTH: url ends_with “.jpg” then prepend request_header “Host” “image-”
```

This rule does not use the term keyword and thus the test condition does not need to reference the request_header “Host” variable. This is a valid rule.

```
RTH: request_header “User-Agent” exists then replace request_header “User-Agent” term “Mozilla/8.0”
```

This is not a valid rule. While it does reference the request_header “User-Agent” variable in the test condition, it does not use a valid test operator as shown in Table 32. As such, there is no “term” to do a replace against. The entire request_header “User-Agent” value is in effect the “term”; therefore the rule should be rewritten without the term keyword as:

```
RTH: request_header “User-Agent” exists then replace request_header “User-Agent” “Mozilla/8.0”
```

This is now valid because the entire User-Agent header value will get overwritten with “`Mozilla/8.0`”.

Note also that when the same variable appears more than once in the test conditions, the linkage between the PAR operation and the test condition is always to the FIRST test condition that meets the criteria shown in Table 32.

For example:

```
RTH: url ends_with “.gif” and url starts_with “/images” then prepend url term “/gif_repository”
```

This rule is probably NOT what the user intended because if the test conditions succeed then the sub-string term “`.gif`” will be prepended with the “`/gif_repository`”, and not the “`/images`” sub-string. This is because the url “term” that was linked to the PAR operator was the first test condition in the rule that matched your criteria. In this case, it was the one that tests for the URL ending with “`.gif`”. To correct this rule, you can simply reverse the ordering of the test conditions:

```
RTH: url starts_with “/images” and url ends_with “.gif” then prepend url term “/gif_repository”
```

Now you can cause a linkage between the PAR operation and the test condition that is looking at the start of the url that would result in the URL being “`/gif_repository/images`”; the correct behavior.

The characters shown in Table 33 are allowed for various PAR strings.

Table 33: Allowable PAR String Variables

Variable	Characters
content	A-Z, a-z, 0-9, space, ',', ':', '`', '!', '@', '#', '$', '%', '^', '&', '(', ')', '=', '+', '[', ']', '{', '}', '|', ';', '"', '<', '>', '?', '~', '.', '/', '_', '-', '*'. Escaped characters are: tab '\t', newline '\n', carriage return '\r', backslash '\\', double quote '\”'. Note that the wildcard character used in test conditions is not escaped as it has no special meaning in a PAR string.
url	a-z, A-Z, 0-9, ',', '~', '%', '.', '/', '_', '-'
query_string variable	a-z, A-Z, 0-9, '=', '+', '&', '?', ',', '~', '%', '.', '/', '_', '-'
header variable	a-z, A-Z, 0-9, '~', '%', ';', ':', '>', '+', '<', '(', ')', '/', '.', ',', '_', '-'

Request Sentry Examples

Table 34 shows some examples of Request Sentry AppRules that should help explain the AppRule grammar.

Table 34: Request Sentry Examples

Example	Description
RS: url length_greater_than “4096” then close_conn FIN and log	
	This example checks to see if the url exceeds 512 bytes and if so, close the connection by sending the client a FIN packet and then logging the result.
RS: any_request_header length_greater_than “2048” then close_conn RST	
	This example checks to see if any headers are longer than 2,048 bytes and if so, immediately closes the connection by sending a RST packet to the client.
RS: request_cookie “session_id” not_exists then redirect “https://www.myserver.com” “/login.cgi”	
	This example determines if the user has a session_id. If they do not, then they are redirected to the SSL /login.cgi URL on the server www.myserver.com.
RS: ssl_version eq “SSLv3” then redirect “https://www.newsite.com” “/login.cgi”	
	This example determines if the user is using SSL Level three, and if he is, redirects the request to the web site: `https://www.newsite.com/login.cgi.` The ssl_version test supports the test operators eq, not_eq, contains, not_contains, ends with, not_ends with, starts with, and not_starts with. The value for ssl_version is case-sensitive, and must be entered in the form: SSLv2, SSLv3, or TLSv1 instead of sslv2, sslv3, or tlsv1.
RS: ssl_ciphersuite eq “DES-CBC3-SHA” then redirect “https://www.newsite.com” “/login.cgi”	
	This example determines if the user is a specific suite of ciphers, and if he is, redirects the request to a different web site. ssl_ciphersuite supports the test operators eq, not_eq, contains, not_contains, ends with, not_ends with, starts with, and not_starts with. The ciphersuites that are allowed are shown in Table 11 on page 186.
RS: ssl_cipher_bits eq “128” then redirect “https://www.newsite.com” “/login.cgi”	
	This example performs a specific action depending on the size of the key. ssl_cipher_bits supports the test operators less_than and greater_than only.
RS: src_ip_1 not_equals "192" and src_ip_2 not_equals "168" and src_ip_3 greater_than "254" and src_ip_4 greater_than "10" and src_ip_4 less_than "1" then close_conn rst	
	In this example, you only allow clients with IP addresses ranging from 192.168.1.1 to 192.168.254.10 to connect. All other clients are rejected abruptly with a TCP 'RST'

Request Translator Examples

Table 35 shows some examples of Request Translator Header AppRules. The last two examples were taken from a sample mappings file.

Table 35: Request Translator Examples

Example	Description
`RTH: url eq “/” then replace url “/pages/top_page.html”`	This example will replace the url from being “/” to becoming “/pages/top_page.html” if the url (i.e., URI) exactly matches the value “/”.
`RTH: url eq “/autos” and request_header “Host” eq “www.myserver.com” then replace url “/” and update_request_header “Host” “autos.myserver.com”`	This example will modify the “Host” header from “www.myserver.com” to “www1.myserver.com” if the “Host” header exactly matches the value “www.myserver.com” and the URI being requested is exactly “/”. Note that in this example, we have placed the action on a separate line. This is perfectly legal since whitespace is ignored.
`RTH: url ends_with “.jsp” then update_request_header “Host” “jspserver.myserver.com”`	This example updates the “Host” header to have the value jspserver.myserver.com if the url ends in “.jsp”.
`RTH: request_header “Host” eq “motorway.dailybulletin.com” then replace url “/Stories/0,1413,250~25660~,00.html”`	This is an example of a URL rewrite.
`RTH: request_header “Host” eq “www.dailynews.com” and url eq “/motorway” then update_request_header “Host” “motorway.dailynews.com” and replace url “/Stories/0,1413,245~25661~,00.html”`	This is an example of a URL rewrite and updating of the Host header.
`RTH: url ends_with “gif” then route_request target_host “192.168.0.2:80”`	This example looks for URLs that end with the file type “gif” and routes those requests to a specific host.
`RTH: url starts_with “/images” then route_request target_host “192.168.0.2:80” “201.201.0.2:80” “198.168.6.2:80”`	This example looks for urls that start with the file path of “/images” and reroutes those requests to one of three specified hosts.
`RTH: src_ip_1 equals "10" and src_ip_2 equals "10" and src_ip_3 equals "0" then reply 302 "http://internal-apps.mycompany.com" "/login"`	This example redirects all incoming requests from the 10.10.0/24 subnet to the login page of an internal application web site. Note that you could have just as easily set up this rule using the traditional 'src_ip' variable like this: RTH: src_ip starts_with "10.10.0" reply 302 "http://internal-apps.mycompany.com" "/login"

Request Retry Examples

Table 36 shows some examples of Request Retry AppRules.

Table 36: Request Retry Examples

Example	Description
`PTH: http_reply_code starts_with “5” then retry_request “3” times same and log`	In this Page Translator Header (PTH) example, if an HTTP request fails with a reply code of 5xx, then retry the request to the same target host (in the cluster where the earlier attempt failed) three more times and log. Other target hosts in the cluster will not be attempted at all.
`PTC: content ci_contains "UNKNOWN" then retry_request “3” times nosame and log`	In this Page Translator Content (PTC) example, the case-insensitive match of the reply content for the word “UNKNOWN” triggers a retry to the subsequent target host in the cluster where the earlier attempt failed. If that attempt fails, move to the next target host in the cluster. Specifying “nosame” means that the initial target host that failed the attempt is never retried. For example, if there are three target hosts (A, B, and C) in the cluster and target host A failed the initial request, target host B is tried first once, then target host C is tried once, then target host B is tried again for a retry count of 3. Note that the target host A was never retried.
`PTC: content ci_contains "UNKNOWN" then retry_request “3" times all and log`	In this Page Translator Content (PTC) example, the case-insensitive match of the reply content for the word “UNKNOWN” triggers a retry to the next target host in the cluster where the earlier attempt failed. If that attempt fails, the retry moves to the subsequent target host in the cluster. Specifying “all” means that the initial target host that failed the attempt is retried when the other target hosts in the cluster have been attempted. For example, if there are two target hosts (A and B) in the cluster and target host A fails the initial request, the target host B is tried first, then target host A, and then target host B again for a retry count of 3.

Request Routing Examples

Table 37 shows two Request Routing examples.

Table 37: Request Routing Examples

Example	Description
`RTH: url ends_with “gif” then route_request target_host “192.168.0.2:80”`	In this Request Translator Header (RTH) example, if an HTTP request is to fetch a page and the URL ends with gif, then the request is served by the target host 192.168.0.2:80.
`RTH: url starts_with “/images” then route_request target_host “192.168.0.2:80” “201.201.0.2:80” “198.168.6.2:80”`	In this Request Translator Header (RTH) example, if the URL requested begins with /images, then the request service is load balanced across the three target hosts specified using Juniper’s Fewest outstanding Requests algorithm. Note: Route Request overrides any sticky load balancing.

Page Translator Examples

Table 38 shows various Page Translator examples

Table 38: Page Translator Examples

Example	Description
PTH: http_reply_code starts with “5” then retry_request same “3” and log	In this Page Translator Header (PTH) example, if an HTTP request fails with a reply code of 5xx, the DX appliance retrys the request to the same target host in the cluster where the earlier attempt failed up to three more times and logs the results. The DX appliance will not try to route the request to other target hosts in the cluster.
PTH: url eq “/” then update_reply_header “Server” “Apache 2.0.47 (Amiga)” “Netscape-Enterprise/4.1” “GWS/2.1”	This Page Translator Header example will essentially update every outgoing request's “Server” header value with one of the three values shown above in a round-robin fashion. This effectively accomplishes the notion of “server cloaking” (or perhaps, server obfuscation from programs or people trying to determine your server type).
PTH: url eq “/” then delete_reply_header “Server”	This would remove the “Server” header from the outgoing reply making it more difficult to tell what kind of origin server is in operation.
PTH: url eq “/login.cgi” and request_cookie “login_challenge” “0” then insert_reply_cookie “login_challenge” “1” “login.myserver.com” “/” secure	This example would update the login_challenge cookie from the value 0 to the value 1 on the outgoing reply. The cookie would only be sent by the client whenever connecting to the server login.myserver.com with an SSL connection. The cookie does not have an expiration date, so it will be discarded by the client when the browser application closes. Note that the insert_reply_cookie action is used instead of update_reply_cookie as we are assuming that the origin server is not sending this cookie for this reply but did so at some time prior.
PTH: url ci_contains "/" then insert_reply_cookie "visit" "yes" ".myserver.com" "/" "3600" secure	This would set a cookie visit to a value yes. The cookie expires 3600 seconds from the time the response is sent to the client.
PTC: content contains “http://*.juniper.net” then replace content term “http://gateway.juniper.net/”	In this Page Translator Content example, the content term “http://*.juniper.net” will be prepended with “http://gateway.juniper.net/” wherever it is found in the response. Note that the search is a case-insensitive “contains” search.
PTH: src_ip starts_with "192.168" and src_ip_3 greater_than "99" and src_ip_3 less_than "105" and src_ip_4 greater_than "0" and src_ip_4 less_than "255" then insert_reply_header "X-Powered-By" "Juniper Web I/O Accelerator"	This trivial example shows how the traditional src_ip test variable can be used in conjunction with the octet-level test conditions to create a rule that tags all replies to clients 192.168.99.1 to 192.168.104.254 with an additional header, "X-Powered-By".
PTC: content ci_contains “</body>” then prepend content term “ Powered by Juniper Networks ”	This example will effectively place a text footer at the end of every HTML page. If you wanted to restrict this to only the home page, you might do something like this:

Table 38: Page Translator Examples

Example	Description
PTC: content ci_contains "</body>" and url eq "/" then prepend content term " Powered by Juniper Networks "	
	You now have two test conditions in operation; one looking for the </body> tag, and one looking for the URL being the home page ("/"). If both are true, then the prepend operation will occur.
PTC: content contains "<%AddBanner%>" then replace content term "<div align=center><imgsrc=http://adserv.doubleclick.net/default_leader.gif alt=\"Click here!\" border=0 width=728 height=90></div>"	
	This example shows how to use the Page Translator Content rule as a special tag replacement mechanism. Wherever the special tag <%AddBanner%> is found, it is replaced with an HTML snippet that displays a banner ad. Note that for speed, the "contains" search is case-sensitive.
PTC: content ci_contains "UNKNOWN" the retry_request nosame "3" and log	
	This Page translator Content example shows that the case-insensitive match in the reply content of the word "unknown" triggers a retry to the subsequent target host in the cluster where the earlier attempt failed.If that attempt fails too, the DX appliance moves on to the next host in the cluster. Specifying "nosame" means that the initial target host that failed the attempt is never retried. For example, if there are three target hosts in the cluster (A, B, and C), and target host A fails the initial request, the DX appliance tries target host B once first, then tries target host C once, and then tries target host B again for a retry count of three. Note that host A was never retried.
PTC: content ci_contains "UNKNOWN" the retry_request all "3" and log	
	This Page translator Content example shows that the case-insensitive match in the reply content of the word "unknown" triggers a retry to the subsequent target host in the cluster where the earlier attempt failed.If that attempt fails too, the DX appliance moves on to the next subsequent target host in the cluster. Specifying "all" means that the initial target host that failed the attempt is retried when all of the other hosts in the cluster have been attempted. For example, if there are two target hosts in the cluster (A and B), and target host A fails the initial request, the DX appliance tries target host B once first, then tries target host A again, and then tries target host B again for a retry count of three.

Limitations/Implications

This section shows the various implications and limitations when using Application Rules.

Application Rules and Latency

Increasing the number of Application Rules will necessarily increase the latency of a request or reply. The amount of added latency varies widely based on the type of rule (its test conditions and actions), and the number of rules. For example, case-insensitive searches take longer than case-sensitive ones. Clusters operating on thousands of rules will execute slower than clusters with just a few rules. In general, higher performance can be obtained by following some general principles:

- Rules are executed essentially in a linear fashion, so place the popular rules near the top of the rule set list.
- Use case-sensitive searches wherever possible.
- The operators `starts_with` and `ends_with` should be used in preference to the contains operator, but use the contains operator if it means fewer individual rules.
- When using the contains operator, use the longest possible string to enable a match, but when using starts_with and ends_with, use the shortest search string.
- Avoid the use of the "*" wildcard where possible.
- Reduce the rule count by using the "`and`" keyword to join test conditions and actions wherever possible.

Displaying Rules

In some cases, the DXSHELL Command Line Interface (CLI) will display the rule that you entered back to you. Because of the way that rules are stored internally, what the system returns to you may not be precisely the way you entered it, although the effect is the same. The variations are as follows:

- A redirect/reply 302 command will have its host information and its URL information combined together as a single string.
- All places where the rule is written with a redirect will be displayed as a reply 302.
- All rule keywords will be in lower case, effectively ignoring the way that you entered it.
- All test operator shorthand notation will be shown in expanded format. For example, "eq" would be displayed as "equals".

User Data Parsing

There is a limit to how comprehensively data entered by a user is parsed for accuracy. The limits are:

- The src_ip (client IP address) value is constrained only to numbers and periods, but there is no check as to the validity of the entry. This is due to the additional complexity of checking a valid entry based against operators that allow partial matching (such as the starts_with operator).
- No checking is performed to ensure that a valid HTTP header name corresponds accurately to either a request header or a reply header. For example, if a user enters "request_header "Server" ci_contains "IIS", this will not be flagged as an error, though in practice it actually is. Future versions of the parser will likely catch such instances.

Test Variable/Action Matching for Prepend/Append/Replace Operations

To keep the internal integrity of an AppRule correct, there is currently a limitation on the relationship between a variable against which a prepend/append/replace operation occurs and the appearance of this variable somewhere in a test condition. For example:

```
PTC: content contains "http://" then replace content term "https://"
```

The variable content has been used in the replace action and also appears as a test condition. Because of this constraint, the following is not legal:

```
PTH: http_request_version equals "1.1" then append reply_header "Warning" ", PTH: enabled"
```

This rule fails because there is no test condition containing the reply_header "Warning" variable. Here is a correction:

```
PTH: http_request_version equals "1.1" and reply_header "Warning" exists then append reply_header
"Warning" ", PTH: enabled"
```

Note that in the second example the reply_header "Warning" variable appears.In related fashion, if a test condition variable is used in a prepend/append/replace operation, but appears multiple times as a test condition, the first instance is the one referred to. This is of importance primarily in the case of operating on the search term, but not the variable's complete value.

For example:

```
RTH: url contains "/images" and url ends_with ".jpg" then prepend url term "/media"
```

Note that the URL term being prepended is the "/images", and not the ".jpg" since the "/images" appears first in the test condition list. If there was some desire to change the second term, then we would do something like this:

```
RTH: url ends_with ".jpg" and url contains "/images" then replace url term ".jpeg"
```

This example will now match the ".jpg" instead of the "/images" search term and replace it with ".jpeg".

Source IP Filtering

When writing rules using the octet-based Source IP variables, remember that it is sometimes necessary to use the opposite test condition to achieve the desired results. For example, if you want to include only the range:

```
192.168.0.1 to 192.168.10.50
```

The first two octets can be satisfied by simply using the "equals" operator to get a valid match. However, the third octet requires that we use the "less_than" operator to get what we want:

```
... src_ip_3 less_than "11" ...
```

This implies that any value less than 11 is okay, which satisfies our example criteria. This type of rule construction is necessary since we do not currently offer compound comparators like "greater than or equal to". The last octet simply requires:

```
... src_ip_4 greater_than "0" and src_ip_4 less_than "51" ...
```

Logging

The Request Sentry rule requires that logging be maintained whenever a rule is flagged with the log action. The logging data is sent to a new log type; "apprule" (as opposed to the "system" log). The log level is alert (ALRT) with the following format:

```
[<timestamp>][S:<source_ip>][D:<vip>][<rule_id>][<request>]
```

where:

- <timestamp> has the current logging format, HH:MM:SS-YYYYMMDD
- <source_ip> is the client's IP address in the format AAA.BBB.CCC.DDD
- <vip> is the VIP of the cluster in the format AAA.BBB.CCC.DDD:PPPP
- <rule_id> is the apprule ID in the format ARID: AAA-BBBB-CCC
- <request> is as much of the request as can be reproduced given the constraints of the maximum logging length for an individual entry. The format is:

```
<method> <url> <protocol_version> <headers>
```

For example:

```
[16:07:22-20030901][S:155.12.33.234][D:19.84.128.12:80][ARID:
023-4555-121][GET /index.html HTTP/1.1 Host:ww]
```

For additional information on the logs, refer to "Syntax of the Log Entries" on page 50.

Configuration Commands

The OverDrive feature is enabled and controlled using the Command Line Interface on the DX appliance. The commands used by the OverDrive feature are:

```
import ruleset <tftp://IP_address/directory>
```

This command is used to migrate a ruleset onto a DX appliance, and it overwrites the current rule if it has the same name. Import also parses the AppRule file for syntax. You can also use the command `<capture file rulesetname>`:

```
set cluster name apprule ruleset <ruleset_filename>
```

This command is used to bind a ruleset to a specific cluster:

```
set cluster name apprule <enabled|disabled>
```

This command is used to enable or disable apprule operation for a cluster:

```
set cluster name apprule limit retrypost <number>
```

This command sets a value that acts as a “high-water mark” for the number of bytes that will be stored for a POST request to be retried. If the POST data exceeds this value, then the data is released and the retry mechanism is disabled for this request. The original request will proceed.

For additional information, refer to “Request Retry, Alerting, and Log (Transaction Assurance) AppRules” on page 289.

This command is used to clear the apprule configuration settings.

```
clear cluster N apprule ruleset <ruleset_filename>
```

Show Configuration Commands

These commands are used to show how the OverDrive feature is configured:

```
show cluster <name> apprule ruleset
show cluster <name> apprule status
show cluster <name> apprule
```

These commands are used to display the AppRule configuration settings. They are used to display AppRule statistics. In each of the statistical DXSHELL commands shown, “M” represents the rule number:

```
show cluster <name> apprule stats [all]
show cluster <name> apprule stats rs [M|all]
show cluster <name> apprule stats rth [M|all]
show cluster <name> apprule stats pth [M|all]
show cluster <name> apprule stats ptc [M|all]
```

The AppRule statistics are cleared when a new ruleset is applied (import ruleset/set server down/set server up).

This command displays all of the limit values:

```
show cluster <name> apprule limit
```

This command displays the retrypost limit value:

```
show cluster <name> apprule limit retrypost
```

This command is used to display AppRule logging information.

```
show log apprule
```

These commands are explained in detail in the *Command Line Reference* manual.

Configuring OverDrive AppRules

To configure a DX appliance to use the OverDrive AppRules, you must first create an AppRule file:

1. Create the AppRule file using the editor of your choice.

2. Enter the AppRules into the file. For example:

```
PTH: HTTP_request_version eq “1.1” and reply_header “Content-Type”
starts_with “text” then insert_reply_header “Vary” “User-Agent,
Accept-Encoding”
PTH: HTTP_request_version eq “1.1” and reply_header “Content-Type”
starts_with “application-x-javascript” then insert_reply_header “Vary”
“User-Agent, Accept-Encoding”
.
```

3. Save the file in pure ASCII format.

4. Import the AppRule file onto the DX appliance using the `import` command.

For example:

```
dx% import ruleset tftp://192.168.40.228/my-ruleset
```

5. The changes will not take effect until you issue a `write` command:

```
dx% write
```

6. Once you have imported the AppRule file into the DX appliance, you must bind the AppRules to a cluster:

```
dx% set cluster <cluster name> apprule ruleset Input_AppRules
(*) dx% set cluster <cluster name> apprule enabled
```

7. To enable the configured AppRules permanently, type:

```
(*) dx% write
```

The steps shown are the preferred method because a syntax check is performed as part of the import process. Another method of entering AppRules is to type them in directly using the Command Line Interface. This method is less desirable because a syntax check is not performed until the AppRule is activated.

1. Create an AppRule file called Input_Apprules, and then enter the desired application rules. End the file with a period (.) as shown below. On a blank line (do not include quotes), enter:

```
dx% capture file Input_AppRules
PTH: HTTP_request_version eq “1.1” and reply_header “Content-Type”
starts_with “text” then insert_reply_header “Vary” “User-Agent,
Accept-Encoding”
PTH: HTTP_request_version eq “1.1” and reply_header “Content-Type”
starts_with “application-x-javascript” then insert_reply_header “Vary”
“User-Agent, Accept-Encoding”
.
```

2. After you have created the AppRules file, you must bind the AppRules to a cluster as shown in Steps 6 and 7 above.

Application Rule Scenarios

This section describes how Application Rules can be used in various scenarios.

Route Request Application Rules

The route_request AppRule add a new action for the Request Translator Header (RTH) to allow users to route a request if an incoming request meets a test condition. You can specify the individual target host, a list of target hosts, or a group of target hosts using these criteria:

- If specified, the RTH rules are evaluated for every incoming request.
- Routing decisions are based upon examining the client request headers to determine which server is appropriate to handle the request.
- If an individual target host is specified for routing requests, load balancing will not be performed. This affects request distribution and means that some hosts may get more traffic than others.
- If a list of target hosts is specified, then "Fewest Outstanding Requests" load balancing is applied across the target hosts within the list.
- Route Requests supersede Sticky load balancing in a cluster.

The suggested request routing syntax is:

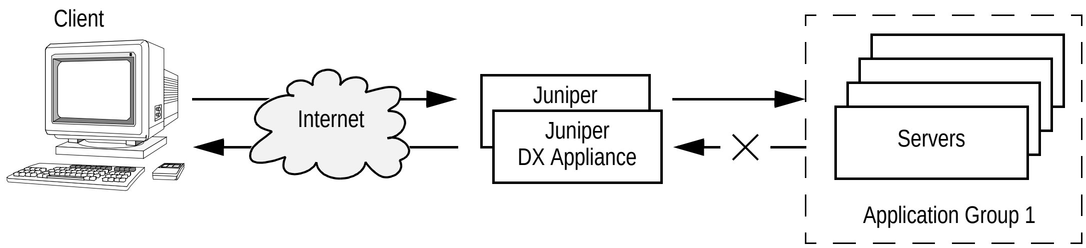
```
RTH: route_request target_host "<ip:port>"
RTH: route_request target_host "<ip1:port1>" ["<ip2:port2>"] …
["<ipN:portN>"]
```

Request Retry, Alerting, and Log (Transaction Assurance) AppRules

Request Retry AppRules allow you to specify retries for unsuccessful HTTP requests. The retries may be based upon the response code for the requests and are testable using DX appliance OverDrive application rules. This feature is used by customers who require a mechanism for adding reliability to HTTP methods.

Lost HTTP responses have a negative impact on not only the end-user who experiences the request failure, but also the business itself as contributions from e-commerce fail to live up to expectations. Retry semantics using the DX appliance OverDrive application rules to the target host help to fix the responses that are lost between the DX appliance and the backend application. This application could be a web, application, database, or an integration server. Refer to Figure 56.

Figure 56: Request Retry Example



The HTTP protocol is a synchronous protocol which requires that a client request completes with a reply (from the origin server, gateway, proxy, or an intermediary) that indicates the success or failure of the request. However, no reliability semantics are built into the HTTP protocol. Since the DX appliance is an intermediary in the HTTP request and response loop, it can initiate a "request retry" for failed requests.

Request Retry AppRules:

- Add a new action for Page Translator Header (PTH) and Page Translator Content (PTC) that allows you to retry a request if the response for a previous request meets a test condition.
- Add the ability to specify maximum number of retry attempts.
- If specified, the PTH or PTC rules are evaluated after every failed attempt.
- Add the ability to specify whether all retries are to one target host, or distributed through the list of target hosts up to the maximum retry limit.
- Add the ability to specify the logging of retry attempts. The logging, if specified, is done after every retry request.
- Makes the AppRule log entries available to an external alerting mechanism (like swatch) for administrator alerts.
- Adds the ability to log failures without retrying the request.

Request Retry syntax is:

```
PTH or PTC: retry_request [same | nosame | all] "number" and log
```

The default is the same target host "number" of times.

You can also set a value that acts as a "high-water mark" for the number of bytes that will be stored for a POST request to be retried by typing the command:

```
dx% set cluster N apprule limit retrypost <number>
```

If the POST data exceeds this value, then the data is released and the retry mechanism is disabled for this request. The original request will proceed.

If a value of zero is specified, then there is no limit imposed on the POST data amount. This is **very dangerous** since it allows a single user to issue a single request

and use all of the resources on the box. The default value is 32768 kBytes. Most POST requests are typically less than 2 kBytes, so there should not be any problems with the default range limits. An upper limit of 100 MBytes is provided for installations that demand maximum flexibility.

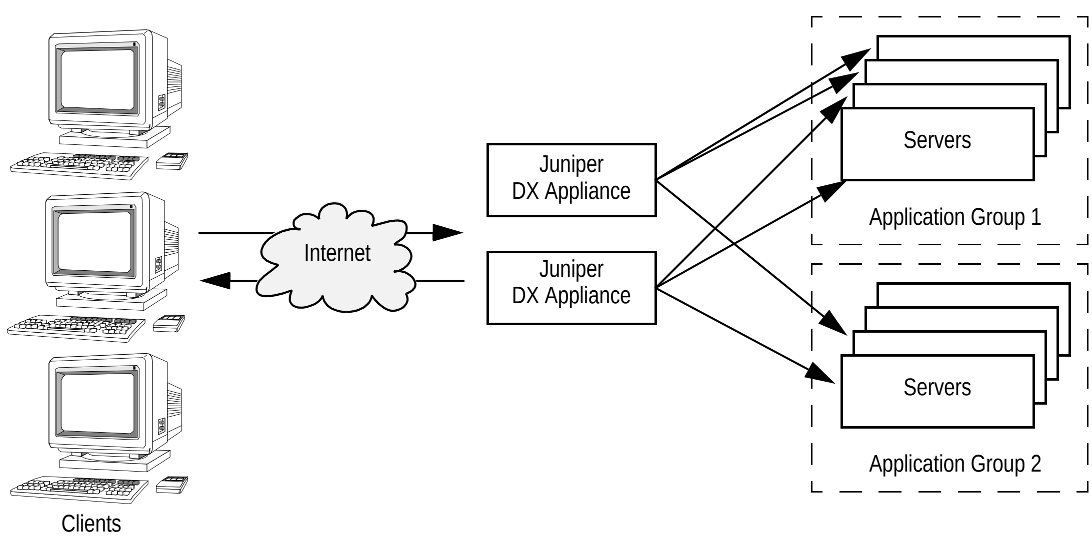
NOTE: For the `retry_request` action to work correctly with Page Translation Content, the factory setting `fcl` must be explicitly enabled (it is disabled by default). Contact your Juniper Administrator.

Request Routing Application Rules

Data centers, web servers, application servers, or integration servers are typically configured to provide distinct services such as payroll, billing, supply-chain integration, etc. A specific server, a list of servers, or even servers collectively referred to as a group, may provide these services. Request routing and dispatch at Layer 7 based upon user-defined information extends the OverDrive functionality to allow users to specify rules that control how requests are routed to user specified targets.

An example of this is a customer care application that routes a client's customer service telephone call to a particular call center (a bank of customer care agents) based upon the caller status (gold, platinum, or executive platinum.) The status can be provided as part of the initial HTTP response, and may be used on all subsequent responses for that session. Refer to Figure 57.

Figure 57: Request Routing Example

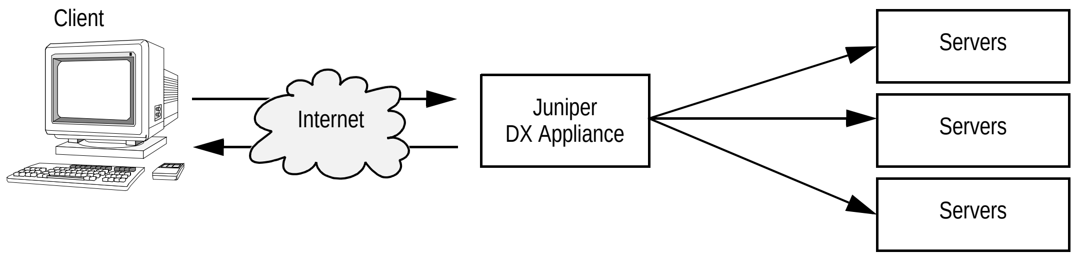


Usage Scenario

An example of routing to a particular target host is a site where target hosts respond with a session ID in the URL that includes an identifier (target host name, IP or custom information) when processing the initial HTTP request from a client. All subsequent requests from that particular client in that session would to be routed to the initial target host (or group of hosts) that handled the initial call. This is a

dynamic processing event and cookies are not used to make route determinations. Refer to Figure 58.

Figure 58: Request Routing Usage Example



Chapter 17

HTTP(S) Authentication

This chapter provides a description of HTTP(S) Authentication for the DX Application Acceleration Platform, discussing the following topics:

- Configuring the Juniper DX Appliance for RSA SecureID on page 307

Overview

Enterprise customers increasingly look to Juniper Networks for user authentication functionality to provide secure access to Enterprise applications and HTTP(S) content. One of the challenges of, and barriers to, migrating from client/server applications to web-based Enterprise applications is security. Authenticating users prior to allowing them access to proprietary HTTP or HTTPS applications is essential.

Because the DX appliance handles all the connections to users, and delivers all of the HTTP and HTTPS traffic to users, it is logical that the DX appliance support user authentication. The authentication methods that Juniper supports are RADIUS, LDAP, and LDAPS. The choice of RADIUS is based on the fact that it a well-entrenched technology that is known and deployed by many of Juniper's customers.

RADIUS can also act as a proxy for several other authentication methods. Some commercial and non-proprietary RADIUS server software packages have the ability to query an external authentication source like an LDAP server or an RSA SecurID server. This gives the DX appliance the ability to move into environments that use other methods of authentication while not having the native support for them. It is

expected that all of the other authentication methods will be supported natively on the DX appliance as needed..

Authentication, Authorization, and Auditing (AAA)

HTTP(S) Authentication fulfills the Authentication function in “AAA” (Authentication, Authorization, and Auditing). Authentication simply identifies a user as who they say they are. The Authorization and Auditing parts of “AAA” are not addressed by this feature.

The DX appliance uses the collected authentication data (i.e., username and password) to satisfy the Authorization part of “AAA” by relaying this information onto the configured authentication server, along with the resource (URL). This provides you with fine-grained control of per-user access to the content fronted by the DX appliance.

The ability of the DX appliance to provide the Authorization part of “AAA” is dependent upon the abilities of the authentication server to provide this service. If the authentication server does have this ability, the DX appliance will pass the collected authentication data along with the user requested resource (i.e., URL) to the authorization server for permissions analysis.

Collecting the Authentication Data

The HTTP specification provides multiple ways to acquire authentication data from the user. The most popular method is to use the WWW-Authenticate and the corresponding Authorization HTTP headers. This method is designed to be used by the origin server. The browser provides the Authorization HTTP header for every request once the user is authenticated. (The header name is misleading when used in the context of “AAA”).

These HTTP headers are not “stackable.” Only one occurrence of each in the request headers is usable. For example, if a web server requires authentication and it finds multiple Authorization HTTP headers in the request, should it (will it) walk through each of them attempting to find the one that works? This would lead to bad security, as well as bad performance.

Additionally, how would a browser “know” to send more than one of these headers with each request? Every time authentication is needed on a request, the browser assumes the previous credentials were not sufficient and prompts for new ones, overwriting the previous one.

Since there is not one solution that is right for all cases, the DX appliance supports all the above methods. The configuration of these options is per-cluster.

The Authorization header is passed on to the server by default. To remove this header you must write an Application Rule. Passing the Authorization header on can be a nice feature in situations where the origin servers are required to perform authentication in addition to the DX appliance and the authentication source is shared. This only pertains in situations where WWW-Authenticate is being used.

Authentication Cache

For every HTTP request coming to an authentication enabled cluster there is a authentication request sent to authentication server. This can overload the authentication servers. Typically authentication servers can handle 100-200 requests/second but the typical HTTP traffic rate is much higher. Authentication cache brings balance by caching successful login attempts and reducing the number of authentication requests forwarded to authentication server.

The data stored in authentication cache is:

- User name and password (input from the user)
- Cluster IP address and port (input from the configuration)
- Password last modified timestamp (input from the authentication server)
- Cache entry expire timestamp (input from the configuration)

The authentication cache size is not configurable. The default size is 1 MByte per multiplexer or 2 MBytes total.

Authentication caching is global and not on a per-cluster basis. The default is that caching is enabled, and authentication caching is persistent across server process restarts.

Authentication Methods

The following authentication methods are supported.

RADIUS

RADIUS is the most popular authentication mechanism deployed. There are multiple commercial, as well as freely available (e.g., open source) offerings available. Each of these servers has slightly different feature sets, but they all share the core RADIUS communication functionality.

The RADIUS authentication protocol is not a perfect one and indeed is not a very secure one, but it is "secure enough." Through the use of shared keys and simple encryption techniques, the data contained in the RADIUS data stream is not visible by sniffing the wire.

RADIUS authentication requests contain the username and password of the person attempting to gain access to the resource, a request identifier, and little else. The RADIUS authentication response contains only the request identifier and the pass/fail/error status code.

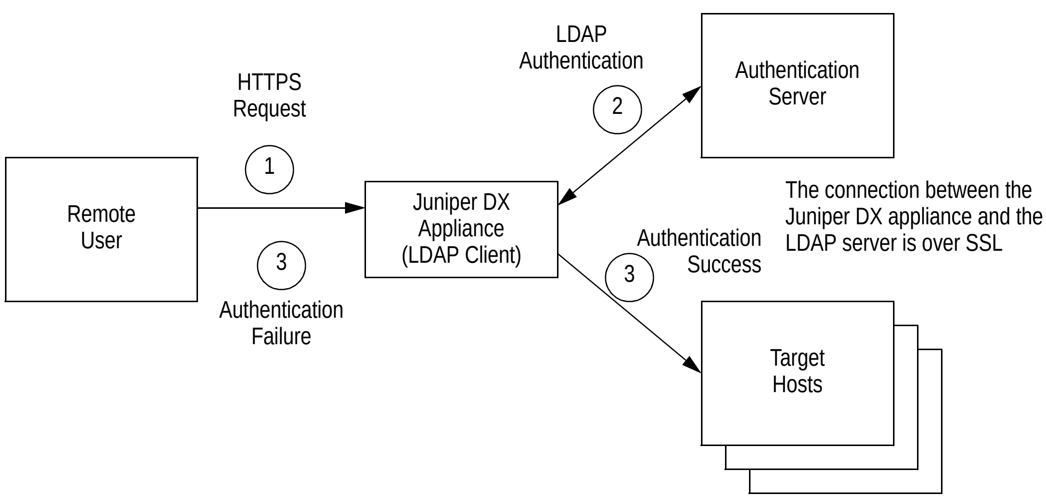
RADIUS servers that are supported include; FreeRADIUS, Cisco, Funk Software's Steel-Belted RADIUS, and possibly others.

LDAP

LDAP has the ability to perform Authentication as well as per-user/per-URL Authorization. When the DX appliance (LDAP client) connects to an LDAP server over SSL, the LDAP v.3 server authenticates itself by sending its server certificate to the DX appliance (refer to item (1) in Figure 59). The DX appliance then needs to determine whether or not the Certificate Authority (CA) who issued the certificate is trusted.

The LDAP server may also request that the client send a certificate to authenticate itself (2). This process is called "certificate-based client authentication" or "mutual authentication". After receiving the client's certificate, the LDAP server determines whether or not the CA who issued the certificate is trusted. If the CA is trusted, the server uses the subject name in the certificate to determine if the client has access rights to perform the requested operation. In order to use SSL, you need a certificate database to hold the CA certificate and (if certificate-based client authentication is used) the client's certificate.

Figure 59: LDAP Authentication



NOTE: The DX appliance acts as an LDAP client, and there can only be one Certificate Authority Authentication Server. This means that in network topologies with multiple clusters, each cluster must address the same Certificate Authority Authentication Server for authentication to work.

Forward Client Certificate

At its simplest level, the DX appliance authenticates a user over a server based upon his username and password. Using "Forward Client Certificate", the authentication has been extended to include authentication scenarios based upon client authentication (refer to Figure 60):

- Authenticate users against a remote server based on the presented client certificate. In this scenario, the DX appliance authenticates the user.
- Accept presented client certificate but does not authenticate locally on the DX appliance. In this scenario, the DX appliance is enabled for client

authentication, but does not authenticate the user. The user is authenticated on the target host.

- Forward the client certificate as an HTTP header. This may apply to both scenarios already listed.

When SSL to the target host is enabled, the client certificate on the DX appliance can be used to establish the target SSL connections. This is another way to get the client certificate to the target host. This has a downside of negating the multiplexing capability of the DX appliance. The following situations can result:

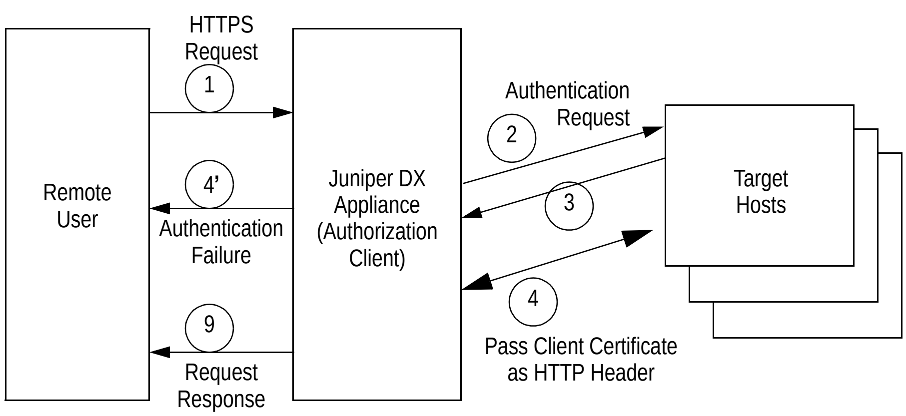
- Clients are authenticated against an database using client certificate.

- Clients are enabled for authentication, but authentication is not done on the DX appliance; the DX appliance forwards the client certificate to the target host for the authentication.

It is possible to reuse the client certificate on the listen side to establish an SSL connection to the target side. This has a downside of disabling the multiplexing capability of the DX appliance.

Currently, when the DX appliance has successfully authenticated the user, the HTTP request is forwarded to the target host. The DX appliance can send the HTTP authorization header, however, the client certificate itself is not forwarded. With Forward Client Certificate the client certificate is forwarded to the target host as part of the HTTP request for further security checks including application level authorization.

Figure 60: Authentication with Forward Client Certificate



Use Case: Health Care Applications

In health care applications, physicians electronically authenticate patient charts with the backend health care system. Each physician is assigned an encrypted digital signature: a secured signature password that cannot be altered or forged by another user. These certificates are used to limit access to patient information and the application logs the access, as required by the Health Insurance Portability and Accountability Act Of 1996. In order to allow Chart One applications to run unaltered when fronted by the DX appliance, the client certificate needs to reach the backend application.

Forward Client Certificate Features

In order to allow downstream applications and devices to validate and authorize the user information, the following requirements are supported:

Client certificate as an HTTP Header

- Allows operators to enable/disable forwarding the client certificate to the target host as an HTTP header
- Supports this capability per-cluster
- Allows operators to define the name of the inserted HTTP header
- Allows operators to choose the format in which the certificate is to be sent. The allowed options are:
  - DER format (X509 base-64 encoded)
  - PEM format.
- Supports client authentication enabled and authenticate against a remote LDAP data store using client certificate
  - Allows operators to enable/disable using client certificate for user authentication
  - Supports this capability per-cluster
  - The username/password is extracted from the client certificate
- Supports client authentication enabled and authenticate locally
- Supports client authentication enabled but don’t authenticate locally
  - Allows operators to enable/disable authentication on the DX appliance. This capability should be done in a way that the DX appliance is able to ask for the client certificate.
  - Once the DX appliance is able to receive the client certificate, the certificate is passed to the target host and DX appliance acts as a passthrough.

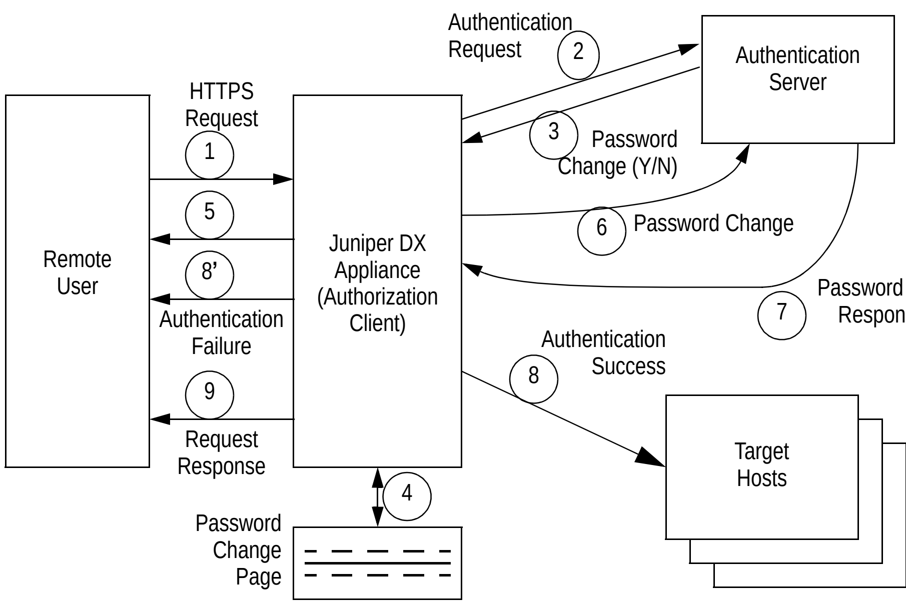
Password Change Request

Cross-platform authentication is a single, centralized password database that can be used to authenticate users on Unix- and Windows-base systems, and other systems such as Macintosh or NetWare.

Microsoft's Active Directory is a directory based authentication system and because LDAP-based authentication is supported on the most recent Microsoft systems (including Windows 2000 and XP), and is also supported on Linux and other Unix systems, LDAP is a good choice for a cross-platform authentication system.

A typical interaction for user authentication when password challenge is employed is shown in Figure 61

Figure 61: Authentication with Password Change Request



1. The user logs by entering his username and password.

2. The DX appliance LDAP client passes the information to the LDAP server (may be the Active Directory).

3. Active Directory (the LDAP Server) sends a password change flag/request to the DX appliance (LDAP Client).

4. The DX appliance invokes the custom password change page with the appropriate fields. The custom page may be a local URL on the DX appliance or a remote URL.

5. The user inputs the requested new password (an old password, if needed). This information is parsed by the DX appliance.

6. The DX appliance forwards the user's response to the Active Directory (LDAP Server).

7. The Active Directory (LDAP Server) may accept or deny the change.

8. The DX appliance forwards the Active Directory (LDAP Server) response to client.

Use Case: On-Line Banking (Password Change on Password Change) Example

An on-line banking company would like to implement a DX appliance as an intelligent web front-end. They would also like to replace the actual infrastructure with a DX appliance. In their existing infrastructure, the user's password validity is three months (12 weeks). After 12 weeks, when the user accesses a web page that requires authentication, the DX appliance should prompt the user with:

```
200: Password Management Page
Change Password Page Username: [User Name]
New Password:
```

At this point, the user should input his new password. The DX appliance must then send this new password to the AD server using LDAP when the user clicks on the [Submit Query] button.

Password Change Requirements

In order to support password change during authentication, the following features are supported:

1. The DX appliance can understand and process the flag "password change required" during an authentication sequence.

2. The DX appliance allows users to define a custom page that will be used to prompt the user with a dialog when a password requires change.

3. The DX appliance can redirect users to a local or remote URL to complete this password dialog on a challenge.

4. This password dialog page can reside on the DX appliance (local) or on another Server (remote.).

5. The DX appliance can allow a URL to identify this challenge page.

The elements of the challenge dialog are specified using DXSHELL. At a minimum, three fields are allowed for the challenge-response/password change page (refer to step 4): The Administration user should be able to specify the name for these fields. For example, when a dialog box for password change needs to be presented, the three fields may be:

- Username
- Old Password
- New Password

6. The DX appliance parses this request. The parsing capability allows the DX appliance to resubmit this request to the authentication server.

7. The challenge-response support in authentication is per-cluster is disabled by default. The write permissions are only allowed for users with the role of Administrator and Security Administrator.

Authentication Commands

Commands used with the authentication feature are.

Set Commands

All the `set` commands need security_administrator or administrator access:

```
set cluster <name> aaa authentication method www
set cluster <name> aaa authentication realm <string>
set cluster <name> aaa authentication response text <string>
set cluster <name> aaa authentication protocol [RADIUS|LDAP]
set cluster <name> aaa authentication [enabled|disabled*]
```

DXSHELL Commands for RADIUS

```
set cluster <name> aaa authentication radius server <name> ip <IP addr>
set cluster <name> aaa authentication radius server <name> port <port number>
set cluster <name> aaa authentication radius server key <shared-key>
set cluster <name> aaa authentication radius server timeout <integer>
set cluster <name> aaa authentication radius server retries <integer>
```

DXSHELL Commands for LDAP

```
set cluster <name> aaa authentication ldap version <integer> /* Default LDAPv3 */
set cluster <name> aaa authentication ldap server <name> ip <IP_addr>
set cluster <name> aaa authentication ldap server <name> port <Port>
set cluster <name> aaa authentication ldap server <name> type <NDS|IPLANET|ADS>
set cluster <name> aaa authentication ldap base-dn <string>
set cluster <name> aaa authentication ldap anonymous [enabled|disabled]
set cluster <name> aaa authentication ldap bind user-dn <string>
set cluster <name> aaa authentication ldap bind password <string>
set cluster <name> aaa authentication ldap uid <string>
set cluster <name> aaa authentication ldap gid <string>
```

Auditing

```
set cluster <name> aaa audit [enabled*|disabled]
set cluster <name> aaa audit level [all|failures]
```

Show Commands

All the `show` commands need security_administrator, administrator, security_operator, or user access:

```
show cluster <name> aaa authentication method
show cluster <name> aaa authentication response
show cluster <name> aaa authentication

show cluster <name> aaa authentication radius server
show cluster <name> aaa authentication radius timeout
show cluster <name> aaa authentication radius retries
show cluster <name> aaa authentication radius key
show cluster <name> aaa authentication radius
show cluster <name> aaa authentication ldap
```

```
show cluster <name> aaa authentication ldap version
show cluster <name> aaa authentication ldap protocol
show cluster <name> aaa authentication ldap server
show cluster <name> aaa authentication ldap base-dn
show cluster <name> aaa authentication ldap bind
show cluster <name> aaa authentication ldap uid
show cluster <name> aaa authentication ldap gid
show cluster <name> aaa authentication ldap anonymous
```

Audit Commands

```
show cluster <name> aaa audit
show cluster <name> aaa
```

Clear Commands

All the `clear` commands need security_administrator or administrator access.

```
clear cluster <name> aaa authentication response text

clear cluster <name> aaa authentication radius server 1 ip
clear cluster <name> aaa authentication radius server 2 ip
clear cluster <name> aaa authentication radius realm
clear cluster <name> aaa authentication radius server key

clear cluster <name> aaa authentication ldap server 1 ip
clear cluster <name> aaa authentication ldap server 2 ip
clear cluster <name> aaa authentication ldap base-dn
clear cluster <name> aaa authentication ldap uid
clear cluster <name> aaa authentication ldap gid
```

Authentication Cache Commands

The following commands are used from DXSHELL to support authentication cache.

Set Authentication Cache Commands

To enable or disable authentication caching, use the command:

```
dx% set cluster <name> aaa authentication cache [enabled* | disabled]
```

To set the maximum age to store an authentication cache entry,
use the command:

```
dx% set cluster <name> aaa authentication cache maxage [maxage]
```

The maxage parameter is in set in minutes, and the default value is 60 minutes.

Show Authentication Cache Commands

To see all of the configuration parameters associated with the authentication cache, type the command:

```
dx% show cluster <name> aaa authentication cache
```

To see all the status of authentication cache, type the command:

```
dx% show cluster <name> aaa authentication cache status
```

To see the maximum time that an authentication cache entry will be stored, type the command:

```
dx% show cluster <name> aaa authentication cache maxage
```

To show the statistics for the authentication cache, type the command:

```
dx% show authentication cache stats
```

Clear Authentication Cache Commands

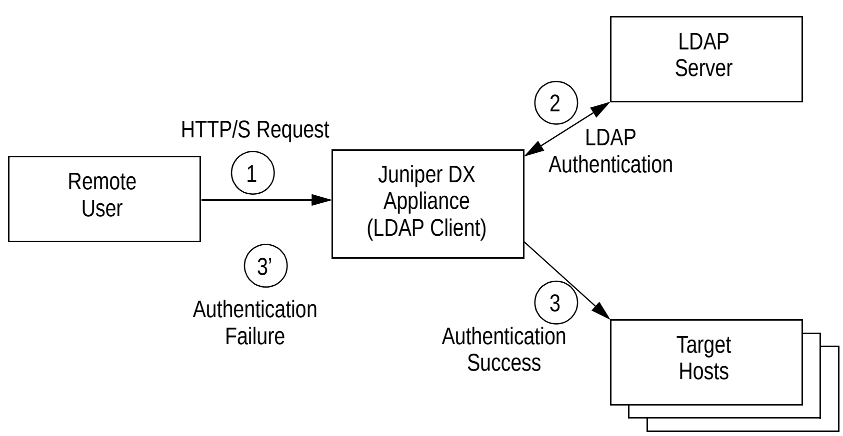
To clear all the configuration parameters associated with the authentication cache, type the command:

```
dx% clear authentication cache
Configuring the Juniper DX Application Acceleration Platform for LDAP and
Active Directory
```

Lightweight Directory Access Protocol (LDAP) is a client-server protocol for accessing directory services. LDAP servers can be used as a focal point for user authentication over the network. LDAP is the industry standard for directory access and is used to provide authentication using the user data stored in an LDAP server.

A typical authentication configuration in the Enterprise configuration includes the following components: LDAP v.3 directory software, one DX appliance, and a remote user, typically a browser connected to a network. The authenticating DX Application Acceleration Platform is a DX appliance that supports LDAP authentication as a client.

Figure 62: LADP Sample Configuration



LDAP System Configuration Overview

Use these general steps to configure your system for LDAP Authentication. The numbers in the steps correspond to the links shown in Figure 62.

1. Refer to the *LDAP v.3 Configuration Guide* to configure your LDAP server. This is available from your vendor for LDAP software. An excellent open source implementation can be downloaded from http://www.openldap.org/. This site also provides a reference guide for configuring OpenLDAP.

2. Configure your DX appliance for LDAP authentication using the configuration steps outlined below.

3. Perform a request to the secured Web or application server using the DX appliance. The DX appliance will prompt for authentication information through a pop-up window on the browser.

Configuring the DX Appliance for LDAP Authentication

Follow these steps to enable LDAP authentication on your DX appliance.

1. Set the cluster parameters for LDAP authentication using these commands:

```
dx% set cluster secure_cluster_001 aaa authentication method www
dx% set cluster secure_cluster_001 aaa authentication realm juniper
dx% set cluster secure_cluster_001 aaa authentication response text
“You are not authorized to access this page.”
dx% set cluster secure_cluster_001 aaa authentication protocol LDAP
```

2. Set the LDAP specific cluster parameters using these commands:

```
dx% set cluster secure_cluster_001 aaa authentication ldap version 3
dx% set cluster secure_cluster_001 aaa authentication ldap server 1 ip 192.168.40.202
dx% set cluster secure_cluster_001 aaa authentication ldap server 2 ip 192.168.40.203
dx% set cluster secure_cluster_001 aaa authentication ldap base-dn
dc=junipernetworks,dc=com
```

The `user-dn` information below is the administrative user (the Manager in this example) for the LDAP directory.

```
dx% set cluster secure_cluster_001 aaa authentication ldap bind user-dn
cn=Manager,dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind password juniper
```

The `uid` information below refers to the column name in the LDAP database that stores the username. By default it is “`uid`” or “`cn`” on most LDAP servers.

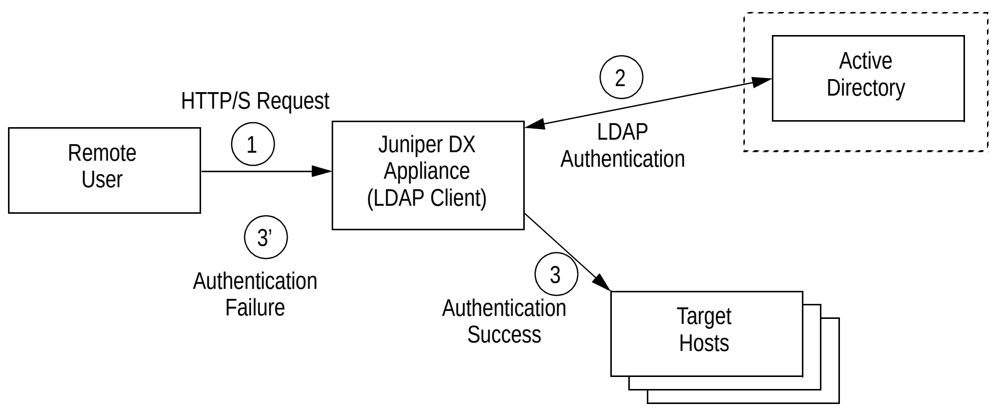
```
dx% set cluster secure_cluster_001 aaa authentication ldap uid cn
dx% set cluster secure_cluster_001 aaa authentication enabled
dx% set cluster secure_cluster_001 target host all enabled
dx% write
```

LDAP and Microsoft Active Directory System Configuration Overview

To use the Microsoft Active Directory as the LDAP server for authentication with the DX appliance, there are a few specific configuration changes that you must make. By default, the Microsoft Active Directory does not permit anonymous LDAP queries. To create LDAP queries or browse the directory, an LDAP client (like the DX appliance) must bind to the LDAP server using the Distinguished Name (DN) of an account that belongs to the administrator group of the Windows system.

Group membership search in the Active Directory is done by enumerating the `memberof` attribute possessed by a given user entry, rather than browsing through the member list in each group. If you change this default behavior to browse each group, you can change the Group Member ID map field from `memberof:member` to `group:member`.

Figure 63: LDAP Authentication with Microsoft Active Directory



Use these general steps to configure your system for LDAP Authentication with Microsoft Active Directory. The numbers in the steps correspond to the links shown in Figure 63.

1. Refer to the configuration guide for the Microsoft Active Directory. This can be accessed at:

   `http://www.microsoft.com/windowsserver2003/technologies/directory/miis/default.mspx`

2. Modify the DX appliance for LDAP authentication to support Active Directory. Refer to the configuration steps outlined below.

3. Perform a request to the secured Web or application server using the DX appliance. The DX appliance will prompt for authentication information through a pop-up window on the browser.

Configuring the DX to Work with Active Directory (via LDAP)

The DX appliance binds by default to the LDAP server before doing any searches, and currently, does not perform group-based queries. Follow the steps to set up Microsoft Active Directory as your LDAP server.

1. Determine the full DN and password for an account in the administrators group. For example, if the Active Directory administrator creates an account in the Users folder of the Active Directory Users and the DNS domain is `juniper.net`, the resulting DN has the following structure:

   ```
   cn=<adminUsername>, cn=users, dc=junipernetworks, dc=com
   ```

2. Set up the information needed to use the Microsoft Active Directory on the DX appliance. Enter the following information in the LDAP settings fields:

   a. Server: The IP address of the machine running the Microsoft Active Directory.

   b. Base Distinguished Name: The domain components of the DN of the account chosen in Step-1. For example: `dc=junipernetworks, dc=com`.

   c. Bind Distinguished Name: The full DN of the account chosen in Step-1. For example: `cn=<adminUsername>, cn=users, dc=junipernetworks, dc=com`.

   d. Bind Password: The password of the account chosen in Step 1.

3. Save the changes.

4. Stop and restart the DX appliance using the commands:

   ```
   dx% set server down
   dx% set server up
   ```

   A sample DX appliance configuration with Active Directory is shown as:

```
dx% set cluster secure_cluster_001 aaa authentication ldap version 3
dx% set cluster ad_secure_cluster_001 aaa authentication ldap server 1 ip <Active Directory IP>
dx% set cluster ad_secure_cluster_001 aaa authentication ldap server type ADS
dx% set cluster ad_secure_cluster_001 aaa authentication ldap base-dn dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind user-dn
cn=<adminUserName>,dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind password juniper
```

   The `uid` information refers to the column name in the Active Directory that stores the username. By default, the Active Directory uses “`anr`”.

```
dx% set cluster secure_cluster_001 aaa authentication ldap uid anr
dx% set cluster secure_cluster_001 aaa authentication enabled
dx% set cluster secure_cluster_001 target host all enabled
dx% write
```

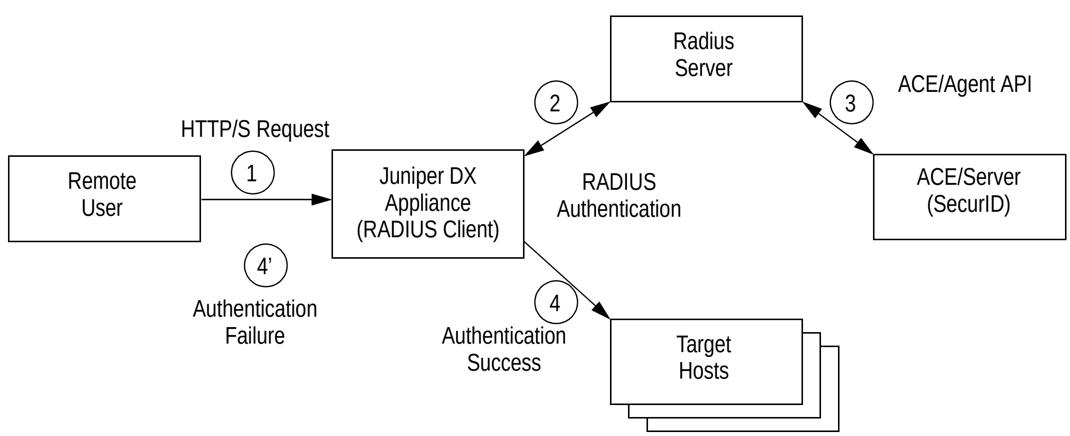
Configuring the Juniper DX Appliance for RSA SecureID

RSA SecurID is one of the more popular token authentication systems. RSA Security's ACE/Server and SecurID solutions provide centralized, two-factor authentication services for enterprise networks and operating systems. This allows only authorized users to gain access to network files, applications, and communications. The DX Application Acceleration Platform supports Enterprise topologies that have standardized on RSA's SecurID.

A typical authentication configuration in the Enterprise configuration includes the following components: the ACE/Server software, one DX appliance, and a remote user, typically a browser connected to a network. The authenticating DX appliance is a DX appliance that supports RADIUS authentication and allows token-based authentication via RADIUS.

A sample configuration is shown Figure 64.

Figure 64: Sample RSA SecurID Configuration



RADIUS System Configuration Overview

Use these general steps to configure your topology for RADIUS authentication:

- Configure the RSA ACE/Server as shown in the ACE/Server configuration guide.
- Refer to RADIUS Server configuration guide to configure your RADIUS Server.
- Configure the DX appliance for RADIUS authentication. Make sure that the secret key matches the one configured on the RADIUS Server. These commands show the configuration needed to enable the DX appliance to work with RADIUS Server.
- Perform a request to the secured Web or application server using the DX appliance. The DX appliance will prompt for authentication information through a popup window on the browser. Note that the RADIUS protocol does not pass the syntax challenge from the authentication server. You will be challenged for a password and not a passcode.

Configuration Steps

The following steps show an example of enabling RADIUS authentication for the DX appliance.

1. Set the cluster parameters for authentication using the commands:

```
dx% set cluster radius_secure_cluster_001 aaa authentication method www
dx% set cluster radius_secure_cluster_001 aaa authentication realm juniper
dx% set cluster radius_secure_cluster_001 aaa authentication response text “You are not authorized
to access this page.”
dx% set cluster radius_secure_cluster_001 aaa authentication protocol RADIUS
```

2. Set the RADIUS specific cluster parameters using the commands:

```
dx% set cluster radius_secure_cluster_001 aaa authentication radius server 1 ip 120.120.120.10
dx% set cluster radius_secure_cluster_001 aaa authentication radius server 1 port 1020
dx% set cluster radius_secure_cluster_001 aaa authentication radius server key <shared-key>
dx% set cluster radius_secure_cluster_001 aaa authentication radius server timeout 20
dx% set cluster radius_secure_cluster_001 aaa authentication radius server retries 3
dx% set cluster radius_secure_cluster_001 aaa authentication enabled
```

3. Save the changes.

4. Stop and restart the DX appliance using the commands:

```
dx% set server down
dx% set server up
```

Chapter 18

Tuning the DX Appliance for Enterprise Applications

This chapter describes tuning the DX Application Acceleration Platform for Enterprise applications, discussing the following topics:

- Target Tuning Tool on page 309
- WebDAV on page 311

Target Tuning Tool

The purpose of Target Tuning is to enable you to easily set up the interaction with target hosts and to properly set up the cluster/system behavior for a custom environment. Target Tuning is a single DXSHELL command that sets a number of configuration variables in an interactive format. The command is:

```
dx% set cluster n target tuning
```

An example of a typical tuning session is shown as follows. The default answer for each of the questions is marked with an asterisk (*):

```
dx% set cluster N target tuning
This will help optimize the communication with the Target Hosts within
this cluster. It will help ensure that functionality is maintained while
providing the most possible benefit.

Please answer the following questions. Enter Control-C at any time to
exit without modification.

1) Please select the Target Application
    1) Other (*)
    2) OWA (Outlook Web Access)
    3) PeopleSoft
    4) Domino 5
    5) Domino 6
    6) JDE OneWorld

Enter Selection: ___

2) Please select the Target Web Server Type
    1) Other (*)
    2) Apache
    3) IIS4
```

```
Enter Selection: ___

3) Is NTLM Authentication used ?
    N) No (*)
    Y) Yes

Enter Selection: ___

You have selected:
        Target Application:  OWA
        Target Web Server:   Other
        NTLM Authentication: Yes

Continue on using these selections ?
    N) No (*)
    Y) Yes

Enter Selection: ___

Tuning based on your selections ...
Done.
```

Question #3 (Is NTLM Authentication used?) will only be presented if the choice for the Target Application does not require connection binding. If connection binding is required by the Application, you cannot unknowingly disable it in question #3.

WebDAV

Web-based Distributed Authoring and Versioning (WebDAV) is a set of extensions to the HTTP protocol that allows users to collaboratively manage and edit files on remote web servers. The primary force driving development of WebDAV is a new generation of programs that “webify” existing Enterprise applications such as Microsoft’s Outlook Web Access (OWA) program.

WebDAV is an in-progress effort of the Internet Engineering Task Force (IETF). The activities in this effort are centralized at the http://webdav.org website. References to all relevant documents, as well as links to applications that use WebDAV can be found here. The WebDAV RFCs and drafts specify a new set of HTTP request methods, response codes, and headers that add to the functionality of HTTP. Additionally, Microsoft has defined an additional set of request methods, response codes, and headers.

The WebDAV methods are only available when you have acquired the appropriate license. Additionally, you have the ability to enable and disable support for these new methods on a per-cluster basis.

Methods

HTTP request methods are divided in two categories: “basic” methods and “enhanced” methods. This was done in order to deliver additional security, as well as to provide fine grained control.

The basic HTTP methods are those that are needed to run any website, intranet, or Enterprise application. The enhanced methods are the rest of the (non-WebDAV) HTTP request methods that are needed in much fewer instances. The basic HTTP methods are always enabled. The enhanced HTTP methods are disabled by default (for security), but can be enabled by any user.

The basic HTTP methods consist of:

- GET
- HEAD
- POST
- PUT

The extended HTTP methods consist of:

- DELETE
- TRACE
- OPTIONS
- CONNECT

Compression of 401 Responses

OWA requires users to login using the standard HTTP WWW-Authenticate mechanism. When the WWW-Authentication header is not present, OWA returns a 401 response code with a relatively large HTML body.

Compression of this HTML content increases the effective compression ratio for the site. In order to support compression of this content, a per-cluster factory option has been added to control compression for 401 responses. It is disabled by default, but it is enabled as part of the recommended WebDAV configuration.

Compression of "text/x-component" MIME Type

OWA delivers content at the beginning of a session with the MIME type, "text/x-component." This content compresses well, and a factory option has been added to enable the compression of this MIME type. It is disabled by default, but it is enabled as part of the recommended WebDAV configuration.

Integration with Application Rules

The new HTTP request methods, headers, and response codes have been added as options to the AppRules. This allows full control of the HTTP traffic by the end user.

Optimization

In order for you to get the maximum value out of the DX Application Acceleration Platform in an OWA deployment, you must enable OWA for the desired cluster. This enables the extended and WebDAV methods, enables connection binding, and enables compression of unauthorized responses and XML and X-component MIME types.

New WebDAV and HTTP Extensions

Table 39 shows the new WebDAV and HTTP extensions that have been added.

Table 39: New WebDAV and HTTP Extensions

New WebDAV and HTTP Extensions			
ACL	CHECKIN	MKRESOURCE	SEARCH
BASELINE-CONTROL	CHECKOUT	MKWORKSPACE	SUBSCRIBE
BCOPY	COPY	MOVE	UNCHECKOUT
BDELETE	LABEL	NOTIFY	UNLOCK
BIND	LOCK	POLL	UNSUBSCRIBE
BMOVE	MERGE	PROPFIND	UPDATE
BPROPFIND	MKACTIVITY	PROPPATCH	VERSION-CONTROL
BPROPMATCH	MKCOL	REPORT	X-MS-EMUMATTS

Table 40 shows the new WebDAV Response Codes that have been added.

Table 40: New WebDAV Response Codes

New Response Codes
102 Processing
207 Multi-Status
422 Unprocessable Entity
423 Locked
424 Failed Dependency
425 Insufficient Space on Resource
506 Loop Detected
507 Insufficient Storage / Cross-Server Binding Forbidden

Table 41 shows the new headers that have been added by the various groups.

Table 41: New Headers

New Headers			
Allow-Rename	Depth	Notification-Type	Subscription-ID
Apply-To-Redirect-Ref	Destination	Ordered	Subscription-Lifetime
Brief	If	Overwrite	Timeout
Call-Back	Label	Position	Transaction
DASL	Lock-Token	Redirect-Ref	
DAV	Notification-Delay	Status-URI	

"Outlook Web Access" (OWA) uses the WEBDAV extensions to the HTTP protocol to provide increased functionality. The DX appliance supports WebDAV extensions to HTTP protocol and allows users to accelerate OWA.

OWA Commands

The following commands support the OWA feature:

```
dx% set cluster <name> owa [enabled|disabled*]
```

This command enables or disables the WebDAV feature:

```
dx% show cluster <name> owa
```

This command shows whether the OWA feature is enabled or disabled, and also shows if the "`child`" commands have been modified by the OWA settings.

For complete information on OWA commands, refer to the *Command Line Reference* manual.

Chapter 19
Performance Monitoring

This chapter describes performance monitoring for the DX Application Acceleration Platform discussing the following topics:

View Juniper Server Statistics

To see real-time statistics for the DX appliance, connect to the DXSHELL command line and type the commands shown in the left column of Table 42.

Table 42: Commands for Viewing Statistics from the DXSHELL Command Line

Command	Definition

`show dashboard`	This command provides a convenient summary of the performance and health of the DX appliance server and the health of clusters and target hosts.
	■ Overall health of the DX appliance's memory, CPU, and network
	■ VIP (Cluster) and Target server health status
	■ DX appliance performance including
	  ■ Bytes the DX appliance sent to clients
	  ■ Connections accepted or refused
	  ■ Requests handled
	  ■ Bytes saved

`show server stats 1`	This refreshes every second and shows: ■ Active TCP sessions, total TCP sessions ■ Active HTTP requests, total HTTP requests ■ Bytes into the DX appliance from the origin web server(s) ■ Bytes the DX appliance sent to clients
`netstat`	Shows the IP addresses and ports of all TCP connections to the DX appliance.
`netstat 1`	Shows the number of packets and bytes being received by the DX appliance from the origin webservers (input) and shows the number of packets and bytes being sent to clients (output). These numbers are updated every second.

The DX appliance also provides a limited selection of real-time performance statistics on the DX appliance Stats page of the WebUI. To view the DX appliance stats page, log in to the WebUI and click on the DX appliance Stats link in the left-hand navigation area. You will see the DX appliance Stats page which provides information about uptime, connections, requests, and bytes in/out.

Capacity Planning

A DXSHELL user has information available to make capacity decisions. The information presented by netstat is very generic and includes information that has spikes. This may unfavorably report system as loaded based on information that is transient. Similarly, the WebUI interface shows Uptime, CPU, Memory, and Network. The WebUI information may change for a few seconds to "red" if the peak value is over a "pre-defined" threshold.

The capacity planning feature allows a user to receive a more informative capacity planning data using the `show capacity` command to show the capacity of the system:

```
dx% show capacity [<seconds>]
```

where `<seconds>` is time intervals for printing the next row. The minimum value for `<seconds>` is 1 and the maximum value is 60. If this argument is missing, only one row of output will be printed and the command will exit.

The displayed values are average over the last minute and the sample interval is one second. This smooths the peaks in the values.

Reporting network usage (including data for `show dashboard`) is limited to Ether 0 and Ether 1. Other interfaces are ignored.

Remote DX Appliance Server Monitoring

Overview

Remote DX appliance Server Monitoring allows the DX appliance to track and aggregate the health and performance statistics for DX appliance in real-world deployments. This information is sent from an DX appliance to the Juniper Service Center every 10 minutes for remote troubleshooting and proactive detection of any issues with the DX appliance deployment.

The collected data is encrypted before it is sent to the Juniper Networks Service Center (service.juniper.net) via HTTP POST over SSL protocol. This service is enabled by default and can be disabled through the command line interface with the command:

```
dx% set server factory svc disabled
```

Information Collected

The following information is collected from all the DX appliances:

- Configuration
- Software version running on the server
- Audit log
- Snapshot of netstat output

- CPU utilization
- Information on internal memory usage, including peak and low-water mark values
- Information on internal processes running on the server
- Information on the TCP connections and HTTP sessions
- Bytes in and bytes out
- System time
- Number of failed transmission attempts

Enabling and Disabling Remote Server Monitoring

Remote Server Monitoring is enabled by default. It can be disabled from the DXSHELL command line.

To disable remote server monitoring, use the command:

```
dx% set server factory svc disabled
```

To turn remote server monitoring back on, use the command:

```
dx% set server factory svc enabled
```

Historical Rates and Statistics

Historical Rates and Statistics allows you to use historical statistics to gauge behavior of your network and data center as well as the ongoing benefit of employing a DX appliance. It allows you to answer questions like: How many bytes have we saved in the past one week, one hour, etc.

The Round Robin Database Mechanism

The Round Robin Database (RRDB) is a mechanism to store time-series data such as network usage, method invocations, etc. It stores the data in a very compact way that does not expand over time, and it can be presented in useful graphs by processing the data to enforce a certain data density.

In order to limit the data acquisition to a finite memory size, the RRDB's store snapshots of information at a pre-set sampling interval or period. These periods are then averaged and rolled into the next higher “bucket.” The round robin, as the name implies, limits the history to the last “n” sampling snapshots in a “bucket” or “category.”

For example, you have a sampling interval of one second and are interested in keeping information about 100 different items for a year. Since there are 60 seconds in minute, 60 minutes in an hour, 24 hours in a day, and so forth, and assuming there are eight bytes of information per statistical field, and 100 fields for historical information, the DX appliance would allocate 149,600 bytes of information.

This implies a database size of:

> (60 sec entries + 60 min entries + 24 hour entries + 30 day entries + 12 month entries + 1 year entry) * 100 items being measured * 8 bytes per item = 149,600 bytes.

This calculation does not include item header information or other housekeeping storage.

At every sampling interval (one second in the example), one entry is made for each statistical item (here 100). When 60 entries are made in the “seconds” bucket, one entry is made into the minute bucket with a value that is computed for the past 60 seconds for that statistic. Then the “second” bucket rolls back to overwriting the oldest value, the first second. The same mechanism is employed for other buckets, i.e., there will be one entry in the hour bucket for every sixty entries in the minute bucket.

Memory Considerations

The Historical Rates and Statistics feature consumes system memory, when you have multiple clusters and target servers, it is subject to some limitations. The limitations imposed by the Flash memory are shown in Table 43 and the limitations imposed by the RAM memory are shown in Table 44.

Table 43: Flash Memory Limitations

Parameter	Limitation
Memory size in Flash for Historical Statistics	4 MByte
Total Fields	738
Instances of each field stored	24 Hours + 31 Days + 12 Months + 1 Year = 68
Memory Requirement for one Cluster	738 * 68 * 8 Bytes = 401472 Bytes
Maximum Clusters supported for Historical Statistics	4 MByte/401472 = 10

Table 44: RAM Memory Limitations

Parameter	Limitation
maximum Memory size reserved for Cluster and Historical Statistics	30 MByte
Total Target Host Fields stored in RAM	572
Total Cluster Fields stored in RAM	140
Instances of each field stored	60 seconds + 60 minutes = 120
Memory Requirements for one Cluster	140 * 120 * 8 Bytes = 134400 Bytes
Memory Requirements for one Target Host	572 * 120 * 8 Bytes = 549120 Bytes
Maximum Target Hosts supported	30 MByte - 134400*10 / 549120 = 54

Description

The Historical Rates and Statistics feature gives you the ability to collect data samples for each statistic item supported by the DX appliance. It adds the ability to specify the number of data samples collected for a statistic item. The sampling interval is fixed at one second and can not be set.

The sampling interval also determines the number of possible entries for the "seconds" table. The default size in the table for seconds data is 60 entries, minutes is 60 entries, hours is 24 entries, days is 31 entries, months is 12 entries, and years is 1 entries.

The sampling interval is not user configurable; the setting has been made at Juniper for optimum results. The trade-offs considered for optimum results are sustained throughput, and performance for connections per second, requests per second, new SSL connections per second, Mbit per second, and simultaneous client and target host connections.

The ability to view statistics is available to a "normal" user. Historical statistics are enabled by default, but can be disabled for a specific cluster.

Historical statistics are written at a one-hour interval to flash. No configuration is allowed.

You can specify a predefined filename to store the historical snapshot data. This is in flash memory. You can then send the data to a remote location using SCP or TFTP.

The sample data is stored as a Comma Separated Value (CSV) file. The format of CSV file is shown in Table 45.

Table 45: Historical Statistics File Format

ClusterName (IP:PORT)	Hour Bucket (1)	Hour Bucket (2)	.... Hour Bucket (24)	Day Bucket (1)	..... Day Bucket (31)	Month Bucket (1)	..... Month Bucket (12)	Year Bucket (1)
10.11.12.13:80	Item1 Item2	Item1 Item2	Item1	Item1 Item2 ...				
10.10.0.1:80								
.....								
.....								
.....								
.....								

Minute and second buckets are not written into the flash. The target host's minute and second historical statistics are stored in memory. Consolidated target host statistics are written to flash at one hour intervals.

You can set high and low watermarks for certain statistics. At a minimum, the following statistics will have this setting:

- Connections/sec
- Requests/sec
- New SSL/sec
- Reused SSL/sec
- Unacked SYNs/sec
- Mbits in/sec
- Mbits out/sec
- 5xx responses/min
- Simultaneous Client connections
- Simultaneous Target connections (requests outstanding)

Statistical Data Items

Historical information is provided for all statistics available in your Juniper DX appliance. You specify these `<stats items>` via Tab-completion. For clarity, the titles of the statistics need to be appended to their `<stats item>` name. A list of `<stats item>` for cluster is provided below for reference.

Methods

- GET
- HEAD
- POST
- PUT
- DELETE
- TRACE
- OPTIONS
- CONNECT
- WebDAV Methods
- Other

Protocols

- HTTP 1.1
- HTTP 1.0
- Other

Browsers

- IE 6.0
- IE 5.5
- IE 5.1
- IE 5.0
- IE 4.x.
- IE Other
- Netscape 4.x
- Netscape 6.0
- Mozilla
- Opera
- Konqueror
- Safari
- None
- Other

Illegal Requests

- Illegal request line too long
- Illegal method
- Illegal 0.9 method
- Illegal POST (no length)
- Illegal POST (length < 0)
- Illegal POST (length = 0)
- Illegal header
- Illegal header line too long
- Illegal PUT (no length)

- Illegal PUT (length < 0)
- Illegal PUT (length = 0)
- Disallowed HTTP Method
- Disallowed WebDAV Method

Responses

- 1xx Responses
  - Summation of all 100 to 102 resp.
- 2xx Responses
  - Summation of all 200 to 207 resp.
- 3xx Responses
  - Summation of all 300 to 307 resp.
- 4xx Responses
  - Summation of all 400 to 425 resp.
- 5xx Responses
  - Summation of all 500 to 507 resp.

Content Types From Servers

- (all types)

Content Bytes From Servers

- (bytes for all types)

Content Bytes To Clients

- (bytes for all types)

SSL

- New Sessions
- Reused Sessions
- Strong Encryption
- Export Encryption
- SSL v2

- SSL v3
- TLS v1
- Other

Connections

- Current Active Server Conns
- Current Idle Server Conns
- Total Server Connections
- Health Checks
- Passed Health Chks (Server OK)
- Failed Health Chks (Server Down)

Client Requests

- Bytes In
- Bytes Out

Target Host

- Bytes In
- Bytes Out

Cluster historical statistics are slightly optimized in order to save memory.

- WebDAV methods are consolidated. WebDAV Method is summation of all types of WebDAV methods.
- Response codes are consolidated.

For reference, a list of `<statistics item>` for target hosts is provided.

Responses

- 1xx Responses
  - (all 1xx responses)
- 2xx Responses
  - (all 2xx responses)
- 3xx Responses
  - (all 3xx responses)

- 4xx Responses
  - (all 4xx responses)
- 5xx Responses
  - (all 5xx responses)

Content Types From Servers

- (all types)

Content Bytes From Servers

- * (bytes for all types)

Content Bytes To Clients

- * (bytes for all types)

SSL

- New Sessions
- Reused Sessions
- Strong Encryption
- Export Encryption
- SSL v2
- SSL v3
- TLS v1
- * Other

Connections

- Current Active Server Connections
- Current Idle Server Connections
- Total Server Connections
- Health Checks
- Passed Health Checks (Server OK)
- Failed Health Checks (Server Down)

Target Host

- Bytes In
- Bytes Out

- Target Decompression Performed
- Target Decompression Failed

Cluster

- Target Decompression Performed
- Target Decompression Failed

Enabling Historical Rates and Statistics

The following DXSHELL commands have been added to support the historical statistical information:

Enabling History

The historical statistics feature is controlled by the license key. To see the license key, type:

```
dx% show license
```

The historical statistics feature is enabled by default, and cannot be disabled server wide, although it can be disabled for specific clusters.

Enabling and Disabling History for a Cluster

To enable or disable historical stats per cluster.

```
dx% set cluster <name> stats history [enabled*|disabled]
```

The default is that history for a cluster is enabled as long as the maximum cluster and maximum target host count is not reached. Currently the sampling rate is set to one second and cannot be changed.

Showing the Cluster Historical Statistics Items

To show the cluster historical statistics items, type the following commands:

```
dx% show cluster 1 stats history [TAB]
http io ssl

dx% show cluster 1 stats history io [TAB]
listen target

dx% show cluster 1 stats history ssl [TAB]
listen target

dx% show cluster 1 stats history http [TAB]
listen target

dx% show cluster 1 stats history http listen [TAB]
browser method reqerr request version

dx% show cluster 1 stats history http listen browser [TAB]
lists of all the browser types.

dx% show cluster 1 stats history http listen browser
<browser-type>[TAB]
second minute hour day month year
```

NOTE:

- A similar format is followed for I/O and SSL listen historical statistics.
- Cluster listen side historical statistics have all the time buckets (second, minute, hour, day, month, and year).
- Cluster target side historical statistics have only hour, day, month, and year.
- Each target host maintains second and minute historical statistics, and they are accumulated for a cluster at every hour interval.
- Target hosts do not maintain hour, day, month, and year historical statistics.

```
dx% show cluster 1 stats history http target [TAB]
bytesin bytesout content responsecode
```

```
dx% show cluster 1 stats history http target bytesin [TAB]
list of all the bytesin stats.
```

```
dx% show cluster 1 stats history http target bytesin
<bytesin-item> [TAB]

hour day month year
```

NOTE:

- A similar format is followed for I/O and SSL target historical statistics.
- Cluster target side historical statistics have only hour, day, month, and year.
- Each target host maintains second and minute historical statistics, and they are accumulated for a cluster at every hour interval.
- Target hosts do not maintain hour, day, month, and year historical statistics.

Showing Target Host Historical Statistics Items

To show the target historical statistics items, type the following commands:

```
dx% show cluster 1 target host ip:port stats history [TAB]
http io ssl
```

```
dx% show cluster 1 target host ip:port stats history http [TAB]
bytesin bytesout content responsecode
```

```
dx% show cluster 1 target host ip:port stats history http
bytesin [TAB]
list of all the bytesin stats
```

```
dx% show cluster 1 target host ip:port stats history http target
bytesin <bytesin-item> [TAB]
second minute
```

NOTE:

- A similar format is also followed for I/O and SSL historical statistics.
- Each target host maintain second and minute historical statistics and they are accumulated for a cluster at every hour interval.

- Target host do not maintain hour, day, month and year historical statistics for space limitations and performance.

Showing Server Historical Statistics Items

To show the server historical statistics items, type the following commands:

```
dx% show server stats history [TAB]
http io ssl

dx% show server stats history io [TAB]
listen target

dx% show server stats history ssl [TAB]
listen target

dx% show server stats history http [TAB]
listen target

dx% show server stats history http listen [TAB]
browser method reqerr request version

dx% show server stats history http listen browser [TAB]
lists of all the browser types.

dx% show server stats history http listen browser <browser-type>
[TAB]
second minute hour day month year
```

NOTE:

- A similar format is also followed for I/O and SSL listen historical statistics.
- Server listen side historical statistics have all the time buckets (second, minute, hour, day, month, and year)

```
dx% show server stats history http target [TAB]
bytesin bytesout content responsecode

dx% show server stats history http target bytesin [TAB]
list of all the bytesin stats

dx% show server stats history http target bytesin <bytesin-item>
[TAB]
hour day month year
```

NOTE:

- A similar format is also followed for I/O and SSL target historical statistics.
- Server target side historical statistics have only hour, day, month and year.
- Server doesn't have target side historical statistics for second and minute bucket.

Clearing Historical Statistics for All Clusters and Target Hosts

To clear the historical statistics items or all clusters and target hosts, type the following command:

```
dx% clear server stats
```

This command clears the historical statistics for all the clusters and target hosts by resetting the counter values to zero.

Clearing Historical Statistics For a Cluster

To clear the historical statistics items for a cluster, type the following command:

```
dx% clear cluster <name> stats
```

This command clears the historical statistics for the cluster, and all the target hosts under that cluster.

DXSHELL Output Example

An example of the `show server stats` command output when viewed from the DXSHELL system console is:

```
dx% show server stats history http listen browser IE6.0 day
-----------------------------------------------------------------------------
Last 31 days                                  IE6.0
-----------------------------------------------------------------------------
                                Absolute Value       |      Delta Value
-----------------------------------------------------------------------------
      Mar    02                       23815162
                                                                2926001
      Mar    01                       20889161
                                                               10189812
      Feb    29                       10699349
                                                                8386042
      Feb    28                        2313307
                                                                2313307
      Feb    27                              0
                                                                      0
      Feb    26                              0
                                                                      0
      Feb    25                              0
                                                                      0
      Feb    24                              0
                                                                      0
      ...
      ...
                                                                      0
      Feb    04                              0
                                                                      0
      Feb    03                              0
```

In this example, the server was started on February 27th. By the end of the day on February 27, the site had received 2313307 hits from users using the Internet Explorer version 6.0 browser. This number became the Absolute Value for February 28th.

By the end of the day on February 28, the site had received 8386042 hits from users using the Internet Explorer version 6.0 browser. This Delta Value (8386042) was added to the previous Absolute Value (2313307) to become the Absolute Value for February 29th (10699349).

All of the `show status` commands use a similar format when executed from the DXSHELL system console.

CSV Export Statistics

The CSV Export feature allows the historical statistics to be saved as a "Comma Separated Value" (CSV) file that can be exported outside of the DX appliance. You can have a separate file for each cluster or a single file for all the clusters. The files are created using either a command from the DXSHELL or from the WebUI. Historical Statistics must be licensed for this feature to be available.

The format of the CSV file with statistics for one cluster is shown in Table 46.

Table 46: Format of the CSV File with Statistics for One Cluster

Item	Hour 1	Hour 2	. . .	Hour 24	Day 1	Day 2	. . .	Day 31	Month 1	Month 2	. . .	Month 12	Year
Item 1													
Item 2													
. . .													
Item N													

The format of the CSV file with stats for all the clusters is shown in Table 47.

Table 47: Format of the CSV File with Statistics for All of the Clusters

Cluster	Item	Hour 1	Hour 2	. . .	Hour 24	Day 1	Day 2	. . .	Day 31	Month 1	Month 2	. . .	Month 12	Year
Cluster 1	Item 1													
Cluster 1	Item 2													
Cluster 1	. . .													
Cluster 1	Item N													
Cluster 2	Item 1													
Cluster 2	Item 2													
Cluster 2	. . .													
Cluster 2	Item N													
Cluster . . .	Item 1													
Cluster . . .	Item 2													
Cluster . . .	. . .													
Cluster . . .	Item N													
Cluster M	Item 1													
Cluster M	Item 2													
Cluster M	. . .													
Cluster M	Item N													

Each of the values is separated by a comma. The items are each of the statistics for which historical statistics are currently available (refer to"Historical Rates and Statistics" on page 318).

Export CSV Statistics Commands

The CSV file can be exported to either a TFTP server of a SCP server. To export the CSV Statistics file, type the command:

```
dx% export cluster <id | all> stats history <dst>
```

This creates the historical statistics file for the given cluster or all clusters and exports it to the specified URL. The file is deleted after the export is complete. This command requires Admin, Network Admin, or Network User administration rights.

The format of the destination `<dst>` is:

```
tftp://tftp_server/filename or
scp://scp_server/filename
```

Double quotes must be used if the filename has spaces:

```
"tftp://tftp_server/dx config"
```

The `<scp_server>` name is a host name or an IP address. The `<filename>` is an absolute path of the file where you would like to export the configuration. The directory specified for the filename must exist.

Exporting CSV Statistics from the WebUI

The CSV file can be exported from the WebUI in one of two ways:

- A link is provided in the Server (DX appliance) statistics page that downloads the Historical Statistics for all the pages. The downloaded data is saved as a file on the client machine.
- A link is provided in the Cluster Stats -> per cluster page that downloads historical statistics for a single cluster. The downloaded data is saved as a file on the client machine.

Advanced Statistics

Overview

Statistics for Input/Output (I/O), HTTP, and SSL are available from the DXSHELL. Statistics can be displayed at the cluster, forwarder and redirector, target host, and physical target levels. Some statistical views can also be narrowed to just the listen or target side.

It is also important to note the difference between a target host and a physical target. A physical target is a web server and a target host is that physical target assigned to a cluster or forwarder. A physical target can be assigned to multiple clusters.

The following are the different categories of statistics available:

- I/O Listen
- I/O Target Host
- I/O Physical Target
- HTTP Listen
- HTTP Target Host
- SSL Listen
- SSL Target Host

I/O Listen Statistics

I/O Listen statistics can be shown at the cluster, forwarder and redirector, and server levels (refer to Table 48).

To display the I/O statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats io
```

To display the I/O statistics for all clusters, type the command:

```
dx% show cluster all stats io
```

To display the I/O statistics for the DX appliance server, type the command:

```
dx% show server stats io
```

Table 48: I/O Listen Statistics

Field	Description
Bytes In (requests from clients) Bytes Out (response to clients)	Cluster or Redirector: Number of bytes at the HTTP level (header or data) received or transmitted by the DX appliance on the listen side. Forwarder: Number of data bytes in TCP packets received or transmitted by the DX appliance on the listen side.
Current Client Connections	Current number of established TCP connections from clients.
Total Client Connections	Total number of TCP connections that have ever been established (SYN, SYN-ACK, ACK) from the clients.
Refused Client Connections	Total number of TCP connections that the DX appliance has accepted from clients and then immediately closed due to resource constraints. A busy message may or may not be sent.

I/O Target Host Statistics

I/O Target Host statistics can be shown at the cluster target host, forwarder target host, cluster, forwarder, and server levels (refer to Table 49).

To display the I/O statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats io
```

To display the I/O statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats io
```

Table 49: I/O Target Host Statistics

Field	Description
Bytes In (responses from servers) Bytes Out (requests to servers)	Cluster: Number of bytes at the HTTP level (header or data) received or transmitted by the DX appliance on the target side. Forwarder: Number of data bytes in TCP packets received or transmitted by the DX appliance on the target side.

I/O Physical Target Statistics

I/O Physical Target statistics can be shown in detail at the cluster target host and forwarder target host levels, or summarized at the cluster, forwarder, and server levels (refer to Table 50).

Table 50: I/O Physical Target Statistics

Field	Description
Current Active Server Connections	Cluster Physical Target: The current number of established TCP connections on the target server side of the DX appliance that are involved in fulfilling a current HTTP request. Forwarder Physical Target: The current number of established TCP connections on the target side of the DX appliance.
Current Idle Server Connections	Cluster Physical Target Only: The current number of established TCP connections on the target server side of the DX appliance that are NOT involved in fulfilling a current HTTP request.
Total Server Connections	Cluster Physical Target: Total number of TCP connections that any cluster has ever established (SYN, SYN-ACK, ACK) to a physical target. Forwarder Physical Target: Total number of TCP connections that any forwarder has ever established (SYN, SYN-ACK, ACK) to a physical target.
Target Status	Cluster Physical Target Level Only (no aggregation): Indicates the connection status to the backend web server (e.g., Up, Layer 7 Down, Transport Protocol Failure, etc.).
Health Check Status	Cluster Physical Target Level Only (no aggregation): Indicates whether health checking is currently enabled or disabled for a physical target.
Passed Health Checks (servers okay)	Cluster Physical Target only: Total number of health checks that have ever passed for a physical target.
Failed Health Checks (servers down)	Cluster Physical Target only: Total number of health checks that have ever failed for a physical target.

HTTP Listen Statistics: Requests from Clients

To display the HTTP statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats http
```

To display the HTTP statistics for all clusters, type the command:

```
dx% show cluster all stats http
```

To display the HTTP statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats http
```

To display the HTTP statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats http
```

To display the HTTP statistics for the DX appliance server, type the command:

```
dx% show server stats http
```

HTTP Listen statistics can be shown at the cluster and server levels.

In Table 51, a "legal" HTTP request is defined as one in which the request line and request headers conform to HTTP standards.

Table 51: HTTP Listen Statistics: Requests from Clients

Field	Description
Requests Active (no reply yet)	Current number of HTTP requests for which the HTTP headers and data are being processed.
Requests Total	Total number of legal AND illegal HTTP requests that have been received by the DX appliance.
Method GET Method HEAD Method POST Method PUT Method DELETE Method TRACE Method OPTIONS Method CONNECT	Total number of legal HTTP requests that have been received with the given HTTP method.
Method PROPFIND Method PROPPATCH Method MKCOL Method COPY Method MOVE Method LOCK Method UNLOCK Method BCOPY Method BDELETE Method BMOVE Method BPROPFIND Method BPROPPATCH Method NOTIFY Method POLL Method SEARCH Method SUBSCRIBE Method UNSUBSCRIBE Method X_MS_ENUMATTS Method VERSION_CONTROL Method REPORT Method CHECKOUT Method CHECKIN Method UNCHECKOUT Method MKWORKSPACE Method UPDATE Method LABEL Method MERGE Method BASELINE_CONTROL Method MKACTIVITY Method BIND Method MKRESOURCE Method ORDERPATCH Method ACL Method Other	Total number of legal HTTP requests that have been received with the given HTTP method. (Continued)

Table 51: HTTP Listen Statistics: Requests from Clients

Field	Description
Version HTTP/1.1 Version HTTP/1.0 Version Other	Total number of legal HTTP requests that have been received with the given HTTP version.
Browser IE 6.0 Browser IE 5.5 Browser IE 5.1 Browser IE 5.0 Browser IE 4.x Browser IE Other Browser Netscape 4 Browser Netscape 6 Browser Mozilla Browser Opera Browser Konquerer Browser Safari Browser None Browser Other	Total number of legal HTTP requests that have been received from the given HTTP browser.
Illegal request line too long Illegal method Illegal 0.9 method Illegal POST (no length) Illegal POST (length < 0) Illegal POST (length = 0) Illegal header Illegal header line too long Illegal PUT (no length) Illegal PUT (length < 0) Illegal PUT (length = 0) Disallowed HTTP Method Disallowed WebDAV Method	Total number of illegal HTTP requests that have been received in the given categories.

HTTP Target Host Statistics

HTTP target host statistics can be shown at the cluster target host, cluster, and server levels (refer to Table 52).

Table 52: HTTP Target Host Statistics

Field	Description
Responses from servers: ** Total 1XX Response Codes ** Response Code 100 Response Code 101 Response Code 102 ** Total 2XX Response Codes ** Response Code 200 Response Code 201 Response Code 202 Response Code 203 Response Code 204 Response Code 205 Response Code 206 Response Code 207 ** Total 3XX Response Codes ** Response Code 300 Response Code 301 Response Code 302 Response Code 303 Response Code 304 Response Code 305 Response Code 306 Response Code 307 ** Total 4XX Response Codes ** Response Code 400 Response Code 401 Response Code 402 Response Code 403 Response Code 404 Response Code 405 Response Code 406 Response Code 407 Response Code 408 Response Code 409 Response Code 410 Response Code 411 Response Code 412 Response Code 413 Response Code 414 Response Code 415 Response Code 416 Response Code 417	Total number of HTTP responses with the given response code values.

Table 52: HTTP Target Host Statistics

Field	Description
Response Code 422 Response Code 423 Response Code 424 Response Code 425 ** Total 5XX Response Codes ** Response Code 500 Response Code 501 Response Code 502 Response Code 503 Response Code 504 Response Code 505 Response Code 506 Response Code 507 Response Code Other	Total number of HTTP responses with the given response code values. (Continued)
Content types from servers: Content GIF Content JPEG Content HTML Content CSS Content XML Content PLAIN Content X-COMPONENT Content JAVASCRIPT Content FLASH Content OCTET-STREAM Content MS-WORD Content MS-EXCEL Content MS-POWERPOINT Content Custom-1 Content Custom-2 Content Custom-3 Content Other	Total number of HTTP responses that contain body content with the given content type.
Content bytes from servers: Bytes In GIF Bytes In JPEG Bytes In HTML Bytes In CSS Bytes In XML Bytes In PLAIN Bytes In X-COMPONENT Bytes In JAVASCRIPT Bytes In HTML Bytes In CSS Bytes In XML Bytes In PLAIN Bytes In X-COMPONENT Bytes In JAVASCRIPT Bytes In FLASH Bytes In OCTET-STREAM Bytes In MS-WORD Bytes In MS-EXCEL Bytes In MS-POWERPOINT Bytes In Custom-1 Bytes In Custom-2 Bytes In Custom-3 Bytes In Other	Total number of HTTP response body bytes received with the given content type (excluding chunk headers) before the DX appliance performs its HTTP-level response body processing and compression.

Table 52: HTTP Target Host Statistics

Field	Description
Content bytes to clients: Bytes Out GIF Bytes Out JPEG Bytes Out HTML Bytes Out CSS Bytes Out XML Bytes Out PLAIN Bytes Out X-COMPONENT Bytes Out JAVASCRIPT Bytes Out FLASH Bytes Out OCTET-STREAM Bytes Out MS-WORD Bytes Out MS-EXCEL Bytes Out MS-POWERPOINT Bytes Out Custom-1 Bytes Out Custom-2 Bytes Out Custom-3 Bytes Out Other	Total number of HTTP response body bytes with the given content type that are remaining after the DX appliance performs its HTTP-level response body processing and compression.
Compressed content bytes to clients: Compressed GIF Compressed JPEG Compressed HTML Compressed CSS Compressed XML Compressed PLAIN Compressed X-COMPONENT Compressed JAVASCRIPT Compressed FLASH Compressed OCTET-STREAM Compressed MS-WORD Compressed MS-EXCEL Compressed MS-POWERPOINT Compressed Custom-1 Compressed Custom-2 Compressed Custom-3 Compressed Other	Total number of HTTP response body bytes with the given content type that were compressed and sent to the client.

SSL Listen Statistics

SSL listen statistics can be shown at the cluster and redirector, and server levels (refer to Table 53).

Table 53: SSL Listen Statistics

Field	Description
Session New	Total number of new SSL sessions that clients have established with the DX appliance.
Sessions Reused	Total number of reused SSL sessions that clients have established to the DX appliance.
Encryption Strong	Total number of SSL sessions with 128-bit or higher level bulk encryption that clients have established with the DX appliance.
Encryption Export	Total number of SSL sessions with lower than 128-bit level bulk encryption that clients have established with the DX appliance.
Version SSLv2 Version SSLv3 Version TLSv1 Version Other	Total number of SSL sessions with the given version that clients have established with the DX appliance.

SSL Target Host Statistics

SSL target host statistics can be shown at the cluster target host, cluster and server levels (refer to Table 54). To display the SSL statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats ssl
```

To display the SSL statistics for all clusters, type the command:

```
dx% show cluster all stats ssl
```

To display the SSL statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats ssl
```

To display the SSL statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats ssl
```

To display the SSL statistics for the DX appliance server, type the command:

```
dx% show server stats ssl
```

Table 54: SSL Target Host Statistics

Field	Description
Session New	Total number of new SSL sessions that the DX appliance has established with a target host.
Sessions Reused	Total number of reused SSL sessions that clients have established to the DX appliance.
Encryption Strong	Total number of SSL sessions with 128-bit or higher level bulk encryption that the DX appliance has established with a target host.
Encryption Export	Total number of SSL sessions with lower than 128-bit level bulk encryption that the DX appliance has established with a target host.
Version SSLv2 Version SSLv3 Version TLSv1 Version Other	Total number of SSL sessions with the given version that the DX appliance has established with a target host.

DXSHELL Commands for Advanced Statistics

In the examples below, the `<name>` field represents cluster and target names or numbers (1, 2, 3, etc.).

Cluster Statistics

To display all the statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats
```

To display all statistics for all clusters, type the command:

```
dx% show cluster all stats
```

To display the I/O statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats io
```

To display the I/O statistics for all clusters, type the command:

```
dx% show cluster all stats io
```

To display the HTTP statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats http
```

To display the HTTP statistics for all clusters, type the command:

```
dx% show cluster all stats http
```

To display the SSL statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats ssl
```

To display the SSL statistics for all clusters, type the command:

```
dx% show cluster all stats ssl
```

Cluster Target Host Statistics

To display the I/O statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats io
```

To display the I/O statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats io
```

To display the HTTP statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats http
```

To display the HTTP statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats http
```

To display the SSL statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats ssl
```

To display the SSL statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats ssl
```

Clearing Cluster Statistics

To clear statistics for a specified cluster, type the command:

```
dx% clear cluster <name> stats
```

To clear statistics for all clusters, type the command:

```
dx% clear cluster all stats
```

Clearing the statistics resets the counter values to 0.

Forwarder Statistics

To display all forwarder statistics, type the command:

```
dx% show forwarder all stats
```

To display all a specific forwarder statistics, type the command:

```
dx% show forwarder <name> stats
```

Forwarder's Target Host Statistics

To display statistics for a specific target host within a forwarder, type the command:

```
dx% show forwarder <name> target host <name> stats
```

To display statistics for all target hosts within a forwarder, type the command:

```
dx% show forwarder <name> target host all stats
```

Clearing Forwarder Statistics

To clear statistics for a specified forwarder, type the command:

```
dx% clear forwarder <name> stats
```

To clear statistics for all forwarders, type the command:

```
dx% clear forwarder all stats
```

Redirector Statistics

To display all the statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats
```

To display all statistics for all redirectors, type the command:

```
dx% show redirector all stats
```

To display the I/O statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats io
```

To display the I/O statistics for all redirectors, type the command:

```
dx% show redirector all stats io
```

To display the SSL statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats ssl
```

To display the SSL statistics for all redirectors, type the command:

```
dx% show redirector all stats ssl
```

Clearing Redirector Statistics

To clear statistics for a specified redirector, type the command:

```
dx% clear redirector <name> stats
```

To clear statistics for all redirectors, type the command:

```
dx% clear redirector all stats
```

DX Appliance Server Statistics

To display all statistics for the DX appliance server, type the command:

```
dx% show server stats
```

To display a one-line summary of DX appliance server statistics updated every n seconds, type the command:

```
dx% show server stats <n>
```

To display the I/O statistics for the DX appliance server, type the command:

```
dx% show server stats io
```

To display the HTTP statistics for the DX appliance server, type the command:

```
dx% show server stats http
```

To display the SSL statistics for the DX appliance server, type the command:

```
dx% show server stats ssl
```

Clearing DX Appliance Server Statistics

To clear statistics for the DX appliance server, type the command:

```
dx% clear server stats
```

Web Log Configuration

Maintaining a Web Log provides vital information for analyzing your web site's traffic. The DX appliance provides the ability to enable a Web Log for a cluster:

```
set cluster <name> weblog [enabled |disabled]
```

When it is enabled, the DX appliance generates a log entry for each HTTP request that it handles.

The DX appliance can be configured to transmit the logs to the Syslog server in one of two ways. The default configuration is Immediate mode, where the DX appliance immediately writes a User Datagram Protocol (UDP) packet containing a web log to the configured Syslog server for each client request. Immediate mode can create a significant amount of extra network activity and does not allow the ability to save logs.

The alternative is Web Log Batch mode. In Web Log Batch mode, web logs are saved on the DX appliance and then copied off in bulk format. For more information, see "Web Log Batch Mode" on page 348.

The user can select the format for the log from one of these five options:

- Common: This is the Apache Common Logging Format (CLF). The information included in the log is:

  ```
  remotehost remotelogname authuser [date] "request" status bytes
  ```

- Combined: This is a modification of CLF (common) format and adds the values of the Referer and User-Agent HTTP headers in quotes:

  ```
  remotehost remotelogname authuser [date] "request" status bytes "Referer"
  "User-Agent"
  ```

- Common_cn: This is a modification of CLF (common) format with the cluster name prepended to the CLF format:

```
clustername remotehost remotelogname authuser [date] "request" status bytes
```

- Combined_cn: This is a modification of the combined format with the cluster name prepended to the combined format:

```
clustername remotehost remotelogname authuser [date] "request" status bytes
"Referer" "User-Agent"
```

- Perf1: This is a proprietary format that allows you to more easily monitor the performance of DX appliance compression and cache. The information included in the log is:

```
remotehost [date] method url version status request-bytes precomp-bytes
postcomp-bytes cachehit
```

- Perf2: This is a proprietary format that allows you to troubleshoot performance problems. The information included in the log is:

```
ip_port from result transactionID T1 T2 T3 T4 Granularity
```

The information fields included in the logs are defined in Table 55.

Table 55: Web Log Field Definitions

Field	Definition
remotehost	The remote hostname (or IP address if the DNS hostname is not available, or if DNSLookup is Off)
remotelogname	The remote logname of the user
authuser	The username with which the user authenticated himself
[date]	The date and time of the request inside brackets ([])
"request"	The request line exactly as it came from the client inside quotes (" ")
status	The HTTP status code returned to the client
bytes	The content-length of the document transferred for response
"referer"	The value of the Referer header inside quotes (" ")
"user-agent"	The value of the User-Agent header inside quotes (" ")
clustername	The name of the cluster that received the request
method	The request method
url	The request URL
version	The request version with the format "HTTP/<major>.<minor>" (without the quotes)
request-bytes	The length of request content-body. This is applicable for POST, PUT, and certain WebDAV requests.
precomp-bytes	The content-length of the response document before compression
postcomp-bytes	The content-length of the response document after compression
cachehit	The number of Juniper cache hits or cache misses
ip_port	This is the IP address and port number for the chosen target server.

Table 55: Web Log Field Definitions

Field	Definition
from	This is the “From” request header.
result	This is the download result. For example, received ack of the last byte of the object.
transactionID	This is the TransactionID response header.
T1	T1 is a timestamp of the request’s arrival completion time on the DX appliance.
T2	T2 is a count of the seconds from T1 until the DX appliance receives the first byte of response from the target server.
T3	T3 is a count of the seconds from T2 until the DX appliance receives the last byte of response from the target server.
T4	T4 is a count of the seconds from T3 until the DX appliance receives the TCP ACK of the last byte of data for this object.
granularity	Granularity is in seconds so the current clock_d can be used for time values.

Web Log Commands

The Web Log is sent out to a Syslog host (server), and the Syslog host must be configured properly before enabling the Web Log feature.

Enabling the Web Log

To configure the Syslog host that will receive the Web Log, type the commands:

```
dx% set cluster <name> weblog syslog host [ip address]
dx% set cluster <name> weblog syslog port [port]
```

The first command sets the host ip address for the Web Log Syslog host. The second command sets the destination TCP port for the Web Log Syslog host. The default port is 514. By default, Web Log messages with the destination Syslog use Local3 as their facility.

To enable the Web Log format, type the command:

```
dx% set cluster <name> weblog format <fmt>
```

This command sets the format, which can be one of common, combined, common_cn, combined_cn, perf1, or perf2.

The delimiter in the Web Log can be set to be either a comma or a space. To set the delimiter, type the command:

```
dx% set cluster <name> weblog delimiter <comma | space>
```

To enable the Web Log feature, type the command:

```
dx% set cluster <name> weblog enabled
```

To disable the Web Log feature, type the command:

```
dx% set cluster <name> weblog disabled
```

Showing the Web Log Configuration

To show the configuration of the Web Log Syslog host, use the commands:

```
dx% show cluster <name> weblog syslog
dx% show cluster <name> weblog syslog host
dx% show cluster <name> weblog syslog port
dx% show cluster <name> weblog syslog format
```

Clearing the Web Log Configuration

To clear the configuration of the Web Log Syslog host, use the command:

```
dx% clear cluster <name> weblog syslog host
```

Web Log Batch Mode

In Web Log Batch mode, Web Logs are saved on the DX appliance and then copied off to an SCP server in bulk format at specified times. The Web Log information that is sent is identical to the information sent in Immediate mode, however it is sent in a batch instead.

The Web Log is stored in compressed format in one of two data stores. The size of these data stores are user-configurable. The second data store is only used when the first data store fills up and the log file cannot be successfully copied. If both data stores fill up and the copy does not succeed, the first data store is purged and filled with new data.

The Web Log is saved on the DX appliance until one of the following events occurs:

- A successful copy is completed
- Both buffers are full
- The DX appliance is rebooted
- You resize the buffer
- You delete or rename the cluster

The Web Log is transmitted securely to the configured Syslog server using Secure Copy (SCP) when:

- You force the DX appliance to send it using a `set cluster <name> weblog batch copy copynow` command
- Maximum buffer size is reached
- User-configurable time is reached (the Alarm)
- Cluster is deleted or renamed

A log message of type EMERG is logged upon a copy failure. You can optionally configure EMERG events to be sent via E-mail.

Before using Web Log Batch mode, you must configure certain items on a per-cluster basis:

- Size of the compressed file to store web logs
- Three alarms that set when to copy (HH:MM in 24 hour format)
- A retry interval if a copy fails (default value: 60 seconds; range: 30-1200 seconds)
- Destination server
- Destination directory
- Secure Copy (SCP) username
- Private key

The DX appliance only supports SSH2 for scp file transfers (SSH1 is not supported). You must upload the private key (RSA or DSA) onto the DX appliance. The private key is captured with a `capture file` command, and the key must not be password protected.

The file name for the copied web logs is set to:

```
<DX hostname>_<cluster_name>_<date><time>.gz
```

The date/time format is YYYYMMDDHHMMSS, and the date and time are based on local time.

Batch Web Logs and Syslog Web Logs are mutually exclusive (only one can be enabled at a time). This feature is enabled with the `set cluster < n > weblog destination` command.

Web Log Batch Commands

You must configure the Web Log Batch feature on a per-cluster basis.

Configuring the Web Log Batch Feature

To determine whether web log entries will be sent to the Syslog server immediately (syslog) or in a batch, type the command:

```
dx% set cluster <name> weblog destination [syslog | batch]
```

The syslog and batch options are mutually exclusive.

To set the size of the compressed file to copy (the size of the two data buffers), type the command:

```
dx% set cluster <name> weblog batch copy size [val in MB]
```

The default value is 10 MBytes, and the range is 1 to 50 MBytes.

To set the times for the Web Log to be transmitted to the configured SCP server, type the commands:

```
dx% set cluster <name> weblog batch copy time 1 [time]
dx% set cluster <name> weblog batch copy time 2 [time]
dx% set cluster <name> weblog batch copy time 3 [time]
```

The format for `[time]` is HH:MM. Up to three times can be configured for each day.

To configure the Web Log to be transmitted to the configured SCP server at periodic intervals, type the command:

```
dx% set cluster <name> weblog batch copy interval <minutes>
```

To force an immediate copy of the Web Logs, type the command:

```
dx% set cluster <name> weblog batch copy copynow
```

To set the retry interval (in seconds) in case of copy failure, type the command:

```
dx% set cluster <name> weblog batch failure retryinterval [val]
```

To set the remote SCP target directory, type the command:

```
dx% set cluster <name> weblog batch scp directory [directory]
```

To set the remote SCP username, type the command:

dx% **set cluster <name> weblog batch scp username [user]**

The private key must be captured using the `capture` command.

To set the (non-password protected) private key, type the command:

dx% **set cluster <name> weblog batch scp keyfile [choose a file]**

To test the connection, type the command:

dx% **set cluster <name> weblog batch scp connecttest**

This copies a one byte test file.

To set the host where the Web Log will be copied, type the command:

dx% **set cluster <name> weblog batch host [server]**

The web log can either be sent to the SCP server in its native format or in compressed form. To enable or disable compression, type the command:

dx% **set cluster <name> weblog batch compression [enable | disable]**

The web logs are compressed in gzip format.

Configuration commands may be executed by users with roles of Administrator and Network Administrator.

Showing the Configuration of the Web Log Batch Feature

To show all of the configuration parameters associated with the Web Log batch feature, type the command:

dx% **show cluster <name> weblog batch**

To show the size of the compressed file to copy (the size of the two data buffers), type the command:

dx% **show cluster <name> weblog batch copy size**

This command also shows the total remaining memory available for weblog batch storage.

To show all three of the times when the Web Log will be transmitted to the configured SCP server, type the command:

dx% **show cluster <name> weblog batch copy time**

To show the interval at which the Web Log will be transmitted to the configured SCP server, type the command:

dx% **show cluster <name> weblog batch copy interval**

To show the retry interval (in seconds) in case of copy failure, type the command:

dx% **show cluster <name> weblog batch failure retryinterval**
To show all of the configuration parameters associated with the remote SCP target directory, type the command:

```
dx% show cluster <name> weblog batch scp
```

To show the remote SCP target directory, type the command:

```
dx% show cluster <name> weblog batch scp directory
```

To show the remote SCP username, type the command:

```
dx% show cluster <name> weblog batch scp username
```

To show the (non-password protected) private key, type the command:

```
dx% show cluster <name> weblog batch scp keyfile
```

To show the host where the Web Log will be copied, type the command:

```
dx% show cluster <name> weblog batch host
```

To show if the Web Log will be sent to the SCP server in compressed form, type the command:

```
dx% show cluster <name> weblog batch compression
```

These commands may be executed by users with roles of Admin, Network Admin, Network Operator.

Clearing the Configuration of the Web Log Batch Feature

To clear the times for the Web Log to be transmitted to the configured SCP server, type the commands:

```
dx% clear cluster <name> weblog batch copy time 1
dx% clear cluster <name> weblog batch copy time 2
dx% clear cluster <name> weblog batch copy time 3
```

To clear the (non-password protected) private key, type the command:

```
dx% clear cluster <name> weblog batch scp keyfile
```

These commands can executed by users with roles of Administrator and Network Administrator.

Chapter 20
Troubleshooting

This chapter describes troubleshooting for the DX Application Acceleration Platform discussing the following topics:

Checking Settings

There are several commands that are useful when troubleshooting your DX appliance. Use these commands to look at your system configuration. For a complete overview of configuration settings of the DX appliance, use the command:

```
dx% show config
```

To obtain basic information about the DX appliance, type the command:

```
dx% show system info
```

For more extensive information about the DX appliance and the environment that it is operating in, type the command:

```
dx% show system debug
```

If you need to call Juniper Networks Technical Support, they will frequently ask for the information displayed by this command. For a complete list of all `show` commands and corresponding `set` commands, refer to the *Command Line Reference* manual or type `show` commands at the DXSHELL prompt.

Troubleshooting

Slow or Degraded Performance

Are Media Settings for Ether 0 Correct?

Mismatched or incorrect media settings will severely impair the performance of the DX appliance. For 1U units, make sure that the media setting for Ether 0 is 100base DX appliance full-duplex. These settings must also match those of the L2 switch port the DX appliance is connected to. **DO NOT USE AUTOSELECT**.

For 2U units with gigabit Ethernet, make sure that the media for Ether 0 and the switch port the DX appliance is connected to are both set to autoselect.

To view media settings, type the command:

```
dx% show ether 0
```

To specify correct media settings for most environments, type the command:

```
dx% set ether 0 media 100baseTX full-duplex
```

Is HTTP 1.1 Enabled on the Target Hosts?

In order for DX appliance to maintain persistent connections with target hosts, the target hosts must be configured to support HTTP 1.1 with keep-alive enabled.

DX Appliance is Not Responding to Requests for Web Content

Verify that the DX Appliance is Serving Web Pages

To make sure that the DX appliance is serving content from the target host, open a browser and enter one of the Virtual IP addresses you set on the DX appliance (remember to enter the port number if it is set to something other than 80). You should see the home page of the target host(s).

NOTE: At LAN speed, the pages may not seem noticeably faster. In part, Web I/O Acceleration addresses the inefficiencies of long-haul and final-mile transfer in order to accelerate page download for people with slower modem and broadband connections. Therefore, acceleration may not be noticeable from within your LAN.

If the DX appliance does not respond, double check your settings and consult the troubleshooting steps that follow.

Are target Hosts Configured and Enabled?

For the DX appliance to serve content, the DX appliance must be configured with one or more clusters populated with target host(s). To view all configured clusters, enter the command:

```
dx% show cluster all
```

Check the output to verify your cluster and target host configuration. Make sure that target hosts are enabled.

Is the DX Appliance Enabled?

If the DX appliance is not responding to requests, check that the DX appliance accelerator is enabled with the command:

```
dx% show server status
```

If the DX appliance is down, bring it up with the command:

```
dx% set server up
```

Be sure to save your change with the command:

```
dx% write
```

Has the DX Appliance Established TCP Connections to the Target Hosts?

Get a list of connections with the command:

```
dx% show netstat
```

Check the output for ESTABLISHED connections to target hosts.

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  10.0.11.20.11910       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.20.11908       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.20.11906       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.20.11904       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.20.11902       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.20.11898       10.0.11.81.80          ESTABLISHED
tcp4       0      0  10.0.11.120.80         *.*                    LISTEN
```

Are the Target Hosts Visible to the DX Appliance?

From the DXSHELL command line, ping one of the target hosts, by typing the command:

```
dx% ping <IP address of the target host>
```

Pinging will stop after five packets on a DX appliance. If the DX appliance can connect to the target host, you should see something similar to this output:

```
PING 192.168.0.102 (192.168.0.102): 56 data bytes
64 bytes from 192.168.0.102: icmp_seq=0 ttl=128 time=0.228 ms
64 bytes from 192.168.0.102: icmp_seq=1 ttl=128 time=0.193 ms
64 bytes from 192.168.0.102: icmp_seq=2 ttl=128 time=0.186 ms
64 bytes from 192.168.0.102: icmp_seq=3 ttl=128 time=0.213 ms
64 bytes from 192.168.0.102: icmp_seq=4 ttl=128 time=0.237 ms
--- 192.168.0.102 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.186/0.207/0.237/0.020 ms
```

Is the DX Appliance Visible from the Target Hosts?

From one of the target hosts, try pinging the DX appliance with the command:

```
dx% ping <IP address of DX appliance ether 0>
```

or

```
dx% ping <Virtual IP address of DX appliance>
```

Pinging will stop after five packets on a DX appliance. You should see something similar to this output:

```
PING 192.168.0.163 (192.168.0.163): 56 data bytes
64 bytes from 192.168.0.163: icmp_seq=0 ttl=255 time=0.219 ms
64 bytes from 192.168.0.163: icmp_seq=1 ttl=255 time=0.174 ms
64 bytes from 192.168.0.163: icmp_seq=2 ttl=255 time=0.174 ms
64 bytes from 192.168.0.163: icmp_seq=3 ttl=255 time=0.187 ms
64 bytes from 192.168.0.163: icmp_seq=4 ttl=255 time=0.181 ms
--- 192.168.0.163 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.172/0.184/0.219/0.016 ms
```

Is DNS Working?

Try pinging a web site from the DX appliance to find out if the DX appliance can resolve the site's domain by typing the command:

```
dx% ping www.google.com
```

If you get the response:

```
ping: cannot resolve www.google.com: Host name lookup failure
```

you can check the DNS settings with the command:

```
dx% show dns
```

You can set the DNS server with the command:

```
dx% set dns server <IP address of DNS server>
```

Is Traffic Flowing Through the DX Appliance?

You can check that the DX appliance is taking in and sending out data with the command:

```
dx% show server stats 2
```

While the stats are refreshing every 2 seconds, try hitting the DX appliance with your web browser. You can tell that the DX appliance is handling traffic by watching the number of Sessions, Requests and Bytes In/Bytes Out increase as the statistics refresh.

```
Uptime: 2 days, 15:51
          Sess                     Req            Bytes In   Bytes Out
    act         tot         act         tot
     3          131          1         1.09K        1.33MB       1.09MB
     3          131          1         1.09K        1.39MB       1.16MB
     3          131          1         1.09K     1.44MB      1.20MB
```

To stop the stats, type the key sequence:

ctrl-c

Cannot Access the WebUI with your Web Browser

Is the WebUI enabled?

Check that the WebUI is enabled with the command:

```
dx% show admin webui status
```

If the WebUI server is down, you can bring it up with the command:

```
dx% set admin webui up
```

Are you Including the Port When You Enter the Address in your Browser?

Check which port the WebUI is listening on with the command:

```
dx% show admin webui port
```

Combine the port with the IP address of ether 0 to form the URL you use to access the WebUI. For example, if the IP of ether 0 was 10.0.11.20 and the admin port was 8090, you would use this URL to access the WebUI:

http://10.0.11.20:8090

NOTE: It is possible to configure WebUI administrator to listen on an IP (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

Cannot Connect to the DXSHELL Command Line with SSH

Is SSH Service Enabled?

Connect though the serial console or the WebUI and check that SSH service is enabled. From the DXSHELL command line, type:

```
dx% show admin ssh
```

If SSH service is down, you can enable it with the command:

```
dx% set admin ssh up
```

Technical Service Dump

The DX appliance can create a complete snapshot of its status intended to accompany support requests to help with remote troubleshooting. All information contained in the dump is available to the user through various commands. The dump is provided as a convenience for expediting the resolution of support requests.

What Information is Collected

- Current configuration
- Data traffic statistics
- System event log information

What Information is not Collected

- Passwords
- SSL keys
- SSL certificates

Creating the Technical Service Dump

Before running tsdump, you will need to configure a few settings that tell the DX appliance what to do with the tsdump file.

1. Choose whether you want to send the tsdump file via e-mail or copy it to your TFTP server. You must choose either e-mail or TFTP; you cannot use both.

   - To send the tsdump via E-mail using your SMTP server:

2. Configure the DX appliance to output tsdumps to E-mail by typing the command:

   dx% **set admin tsdump transport smtp**

3. Specify an SMTP server that the DX appliance can use to relay E-mail by typing the command:

   dx% **set admin email server <IP address of SMTP server>**

4. Set a name for the tsdump file by typing the command:

   dx% **set admin tsdump filename <filename>**

NOTE: On some operating systems, including most UNIX-like systems, TFTP upload requires an existing, writable file with the same filename on the remote host.

---

5. Specify the from E-mail address that should appear in the E-mail by typing the command:

```
dx% set admin email from <e-mail address>
```

6. Specify up to two different E-mail addresses to send the tsdump to by typing the command:

```
dx% set admin tsdump mailto1 <e-mail address>
dx% set admin tsdump mailto2 <e-mail address>
```

7. To copy the tsdump file to your TFTP server, tell the DX appliance to output tsdumps to a TFTP server by typing the command:

```
dx% set admin tsdump transport tftp
```

8. Tell the DX appliance which TFTP server to use with the command:

```
dx% set admin tftp server <IP address of TFTP server>
```

9. Set a name for the tsdump file by typing the command:

```
dx% set admin tsdump filename <filename>
```

NOTE: On some operating systems, including most UNIX-like systems, TFTP upload requires an existing, writable file with the same filename on the remote host.

---

10. Finally, after completing the required tsdump settings, create and send the tsdump with the command:

```
dx% tsdump
```

Using tcpdump to Get a Detailed Report of Network Activity

The program tcpdump provides a detailed report of network activity that can be useful for troubleshooting.

Using the tcpdump Utility

Before running tcpdump you will need to configure a few settings that tell the DX appliance what to do with the tcpdump file.

1. Choose whether you want to send the tcpdump file via e-mail or copy it to your TFTP server. You must choose either E-Mail or TFTP; you cannot use both.

   - To send the tcpdump via E-Mail using your SMTP server, configure the DX appliance to output tcpdumps to E-Mail by typing the command:

   ```
   dx% set admin tcpdump smtp
   ```

2. Specify an SMTP server that the DX appliance can use to relay e-mail by typing the command:

   ```
   dx% set admin email server <IP address of SMTP server>
   ```

3. Specify the “from” E-Mail address that should appear in the E-mail by typing the command:

   ```
   dx% set admin email from <e-mail address>
   ```

4. Specify up to two different E-Mail addresses to send the tcpdump to by typing the commands:

   ```
   dx% set admin tcpdump mailto1 <e-mail address>
   dx% set admin tcpdump mailto2 <e-mail address>
   ```

5. Tell the DX appliance output tcpdumps to a TFTP server by typing the command:

   ```
   dx% set admin tcpdump tftp
   ```

6. Tell the DX appliance which TFTP server to use by typing the command:

   ```
   dx% set admin tftp server <IP address of TFTP server>
   ```

7. Set a name for the tcpdump file by typing the command:

   ```
   dx% set admin tcpdump filename <filename>
   ```

NOTE: On some operating systems, including most UNIX-like systems, TFTP upload requires an existing, writable file with the same filename on the remote host.

8. Create the tcpdump file with the command:

   ```
   dx% tcpdump
   ```

9. Copy the tcpdump file off of the DX appliance via the method specified in step 2 (TFTP or E-mail) by typing the command:

```
dx% copy tcpdump
```

Viewing a tcpdump File on the DX Appliance

After creating a tcpdump file on the DX appliance you can immediately view the output with the command:

```
dx% show tcpdump
```

Viewing a tcpdump Outside the DX Appliance

NOTE: If you are running release 2.3 or later, tcpdump output is already in binary format and you can skip to STEP 2. You can see what release you are running with the command `show version.`

---

1. For release 2.2.x, decode the base64-encoded tcpdump file.

2. For decoding base64-encoded files, you can use the `uudecode` command on most UNIX machines. You can also download a base64-decoding utility, `base64.exe`, for Windows-based computers from the Juniper Networks Technical Support site.

   Use `uudecode` to decode the tcpdump file by typing the command:

```
dx% uudecode <tcpdump to decode> <filename of new decoded tcpdump>
```

3. To use `base64.exe` to decode the tcpdump file on a Windows-based computer, select "Run..." from the Start menu and enter the command:

```
dx% <path to base64.exe>base64.exe -d <tcpdump to decode> <filename of new decoded
tcpdump>
```

   For example:

```
C:\WINNT\Profiles\administrator\Desktop\base64.exe -d tcpdump
tcpdump_decoded
```

   Note that if you copy `base64.exe` to a directory in your Windows $PATH, you can omit the path to `base64.exe` and simply use the command:

```
dx% base64 -d tcpdump tcpdump_decoded
```

4. Type `PATH` at the Windows command prompt to see directories in your PATH. PATH typically includes `C:\WINNT` and `C:\WINNT\system32`

5. Once the file is decoded, you can view it using a standard tcpdump utility with the command:

```
dx% tcpdump -r <name of decoded tcpdump file>
```

   You can also use a protocol analyzer such as Ethereal to view the decoded tcpdump.

Appendix A
Glossary

Table 56: Glossary

Term	Definition
Active-Active	An Active-Active configuration is a two-DX appliance configuration where both DXs are actively processing client traffic and load-balancing the client requests. One of the DX appliances is the token “Master” and if the Master DX fails, the remaining DX appliance takes up the master role to take and redistribute the requests from clients.
Active-Standby	An Active-Standby configuration is a two-DX appliance configuration where one DX processes client traffic and load-balances the client requests (the active unit) while the other (standby) unit listens to the active unit’s heartbeat and waits to take over as the active unit in case of the active unit’s failure.
ActiveN	An ActiveN configuration is an extension of the two-DX Active-Active configuration where up to 64 DXs are actively processing client traffic and load-balancing the client requests. One of the DXs is the token “Master” and if the Master DX fails, one of the remaining DXs takes up the Master role to take and redistribute the requests from clients.
Blade	A blade is a DX that has been configured as part of a Group.
“Busy” Redirect	If the Target web server responds with a “Busy” error, the Web I/O Accelerator will serve the page specified by this URL instead.
Certfile	Certification file for SSL traffic.
Cipher	Cryptographic algorithm for a server and client to authenticate each other, transmit certificates, and establish session keys.
Ciphersuite	A set of ciphers.
Cluster	A cluster is a collection of web servers that are all configured to serve the same content for a single web site (refer to “Web Cluster” on page 75), and to be accelerated by the DX. The DX listens for incoming web traffic on a specific virtual IP address and port, distributes it over the target hosts (web servers) in the cluster and then accelerates the outgoing web traffic. Typically all the web servers in a particular cluster serve identical content; that is, each cluster usually represents a distinct website or property.
Convert302protocol	Converts the 302 responses from HTTP to HTTPS or from HTTPS to HTTP.
Customiplogheader	A special header to annotate the log; showing the session that is being logged in an easily identifiable way.
Custom Header	This is custom HTTP header that will be added with the client’s origin IP to the client's request.
Default Route	Also known as the “Gateway,” this is the IP address of the machine the Web I/O Accelerator talks with in order to access the outside world.

Table 56: Glossary

Term	Definition
Direct Server Return (DSR)	Direct Server Return is a configuration where incoming client packets are sent to the Layer 4 Switch, but outbound target blade packets are sent directly back to the client. This reduces the outgoing traffic channeled through a load balancer by allowing web servers to send their HTTP responses directly back to the requesting client without passing back through the load balancer. Enable this option on the Web I/O Accelerator if the target web servers are configured to use DSR. For additional information, refer to "Connection Handling" on page 381.
DNS Domain	Also known as the Domain Suffix; this will be used to resolve unqualified host names.
DNS Nameserver	The IP address of the primary name server for the Web I/O Accelerator. This is the machine the Web I/O Accelerator queries to resolve host names into IP addresses.
Ethernet 0 (ether0)	This is the primary ethernet port of the Web I/O Accelerator and the interface through which web traffic travels.
Ethernet 1 (ether1)	Also known as the "Heartbeat" port, Ether 1 is used to communicate with a second Web I/O Accelerator configured as a cold-standby fail-over unit.
Farm	A farm (also called a Server Farm) is a larger collection web servers that are configured to serve either a single a several web sites (refer to "Web Farm" on page 76). Within the farm, the servers are frequently configured in clusters, each serving a single web site.
Failover	A process where two or more DXs monitor each other's health, and if one DX appliance fails, another one takes over the processing of new requests. This specifies whether or not the Web I/O Accelerator should act as a cold-standby fail-over unit for another Web I/O Accelerator on the network. NOTE: Both the active and the stand-by DXs should have this option enabled, and both units should have the same Virtual IP settings
Forwarder	A Forwarder is a mechanism for forwarding traffic on to a set of servers. It listens for incoming traffic on a specific virtual IP address and port and distributes it over the target hosts. Unlike a cluster, a forwarder blindly forwards incoming traffic on to its target hosts. These typically are not web servers, and the forwarder does not attempt to accelerate the outgoing traffic. This is for non-HTTP traffic; the forwarder simply passes the traffic through without examining it.
Group	A group is a homogenous collection of Juniper DXs (also known as "blades"), that is being serviced by a Layer 4 Switch. Any of the DX appliances is capable of servicing a request. Within the Layer 4 Switch, the concept of a group is similar to the concept of a cluster that exists within the DX appliance.
Hostname	The fully qualified DNS name for the Web I/O Accelerator.
Instant Redirect	Instant Redirect is a mechanism where the DX monitors the health of the target hosts in a cluster, and diverts traffic from a cluster where all target hosts are down (i.e., a "dead" cluster) to an active cluster somewhere else in the network (world).
Keyfile	Key file for SSL traffic.
Keypass	Password for SSL key.

Table 56: Glossary

Term	Definition
Layer 4 Switch	A Layer 4 Switch (L4S) is a packet-based switch based on the OSI "transport" layer. Layer 4 switches identify which application protocols (HTTP, SNTP, FTP, and so forth) are included with each packet and use this information to hand off the packet to the appropriate blade or cluster. Layer 4 switches alleviate server load by balancing traffic across a group of DX appliances (blades) or a cluster of servers based upon individual session information and status. When an L4S is placed in front of cluster of servers running a particular application, and a client makes a request for that application, the switch determines which server should handle the request, often based upon current server loads. Once the forwarding decision is made, the switch binds that session to a particular server.
Layer 7 Health Checking	Checks whether the target hosts are available by periodically sending an HTTP request to a specific URL on the target hosts.
Layer 7 Health Check Request Interval	The number of seconds separating each health check request sent to the the target hosts. The valid range of values is 1 - 60 seconds.
Layer 7 Health Check Request URL Path	The URL path that is requested on a target host with each health check. The URL path must begin with a slash "/".
Layer 7 Health Check Retry Threshold	The number of times a health check must fail before the target host is considered unavailable. The valid range of values is 1 - 20.
Layer 7 Health Check Resume Threshold	The number of times a health check must succeed before the target host is considered available. The valid range of values is 1 - 20.
Layer 7 Health Check Status Code	The HTTP response status code expected from a target host in response to a health check. For typical use, the status code should be set to 200.
Layer 7 Health Check Page Size	The page size expected from a target host in response to a health check. This is the number of bytes in the body of the HTTP response, as it would be indicated in an HTTP Content-Length header. This is an optional setting; to disable this setting, use the value -1.
Layer 7 Health Check Expect String	A string expected to appear somewhere in the HTTP response given to a health check. The expect string is searched for in the non-header portion of the HTTP response. It is case-sensitive and must be enclosed in double-quotes if there is whitespace in the string. The maximum length of the string is 64 bytes. This setting only applies to health check responses with the following MIME types: text/html, text/css, text/plain and text/xml. This is an optional setting.
Listen Port	The port on which the Web I/O Accelerator listens for incoming web traffic; it is typically set to 80.
Listen IP Address	Refer to Virtual IP Address.
Listen IP Netmask	Refer to Virtual IP Netmask.
Log Host	The IP address of the server to which the Web I/O Accelerator will be sending logging data.
Logging	Turns logging on or off. Remember that logging always exacts a performance penalty.
MAC Address	The Media Access Controller (MAC) address is a hardware address that uniquely identifies each node of a network. This address is represented in the form of six hexadecimal numbers, typically separated with colons (For example: 20:4A:3E:44:00:22). This should not be confused with the IP Address.

Table 56: Glossary

Term	Definition
Media	Media is the mode in which an Ethernet interface (Ether 0 and Ether 1) operates.
MTU	Maximum Transmission Unit (MTU) is the largest number of bytes of “payload” data a frame can carry, not counting the frame's header and trailer. The MTU should be set to 1500 for Ethernet. DO NOT change this value unless your switch and network are configured to work with a different MTU.
Netmask	A mask to filter out addresses that should not access the device.
NTP	Network Time Protocol. Specifies whether or not the Web I/O Accelerator should listen for your NTP server.
RADIUS	Remote Authentication Dial In User Service
Redirector	A Redirector is mechanism for redirecting requests to a single web server. It listens for incoming web requests on a specific virtual IP address and port and redirects the client to that web server. Unlike a cluster, a redirector does not allow web traffic to pass through the Web I/O Accelerator. Instead, for every web request a redirector receives, the redirector sends the client back a redirect URL and forces it to resend its HTTP request to that URL.
Redirector Host	The host portion of the redirect URL sent by the redirector. That is, this is the web server to which the client should be redirected. The redirector host may be specified as either a hostname or an IP address.
Redirector Port	The port portion of the redirect URL sent by the redirector.
Redirector Protocol	The protocol portion of the redirect URL sent by the redirector. Valid values are HTTP and HTTPS.
Redirector URL Method	The manner by which the redirector specifies the path portion of the redirect URL. If the request method is selected, then the redirector will construct the redirect URL using the same URL path as the original request. If the custom method is selected, then the redirector will construct the redirect URL using a custom URL path. You must specify a custom URL path if the custom method is selected, and the custom URL path must begin with a slash '/'. For instance, if the request method is selected and the redirector receives a request for a page at '/path/page.html', then the redirect URL will look something like 'http://my.redirect.host/path/page.html'. However, if the custom method is selected and the custom URL path is set to '/custom/script.cgi?a=b', then the redirect URL will look something like 'http://my.redirect.host/custom/script.cgi?a=b' for any request received by the redirector.
RMMP	The Redundancy Multicast Messaging Protocol (RMMP) is a mechanism where the active Layer 4 Switch sends health messages that the other Layer 4 Switch receives. This messaging protocol enables health checking between DX appliances. If a certain number of health messages are not received within a time window, the second Layer 4 Switch takes over the processing of new requests.
Route (Default)	Also known as the “Gateway”. This is the IP address of the machine the Web I/O Accelerator talks with in order to access the outside world.
Server	Web I/O Accelerator service.
Server Load Balancer	A Server Load balancer (SLB) distributes service requests across a group of target hosts, based on their availability to service requests.

Table 56: Glossary

Term	Definition
SSL	Secure Sockets Layer (SSL) is a protocol that defines a way for two network devices to communicate securely. You can enable SSL on the listen side to communicate with clients securely. You can enable SSL on the target side to communicate with the target hosts securely
SSL Protocol Version	There are three versions of SSL protocol: SSL version 1 (SSLv1), SSL version 2 (SSLv2) and Transport Layer Security version 1 (TLSv1). There are four SSL protocol modes in which the Web I/O Accelerator can operate: sslv2: Use SSLv2 only sslv3: Use SSLv3 only sslv23: Use SSLv2, SSLv3 and TLSv1 tslv1: Use TLSv1 only
SSL Ciphersuite	A collection of cryptographic algorithms used by two network devices to authenticate one another, transmit certificates and establish session keys. There are four categories of cipher suites used by the DX: all: Allow all supported SSL ciphersuites common: Allow only the fastest ciphersuites from both the strong and export groups export: Allow only the low security ciphersuites suitable for export strong: Allow only the highest security ciphersuites suitable for use in the U.S.A.
SSL Certfile	The certificate file used when establishing SSL communication.
SSL Keyfile	The key file used when establishing SSL communication.
SSL Keypass	The password for the SSL Keyfile.
Sticky	Ties a client to a server via the cookie or the client's IP address.
Sticky Load Balancing	A method of load balancing that binds a client to a server via a cookie or the client's IP address. It ensures that all subsequent requests made by a client are directed to the same server that handled the initial request.
Target Host:Port	This is the IP address and accompanying port of the web server that the Web I/O Accelerator will accelerate. Depending upon the Web I/O Accelerator model, you may be able to enter IP addresses and ports for up to eight Target Hosts.
Target Name	This is the fully-qualified host name which clients use to reach your website or the servers you are accelerating.
Web I/O Accelerator Statistics	The following Web I/O Accelerator Statistics are available: Uptime: The elapsed time since the Web I/O Accelerator was turned on. Sessions (active/total): The number of TCP sessions that the Web I/O Accelerator has handled. Requests (active/total): The number of HTTP requests the Web I/O Accelerator has received. Bytes (in/out): The total amount, in bytes, of data the Web I/O Accelerator has received from target hosts, and the total amount of data that the Web I/O Accelerator has sent out to clients.
Virtual IP Address	This is the IP address to which all incoming web traffic should be routed. It should be different from the IP address(es) you specified on the Network Settings page.
Virtual IP Netmask	The proper subnet mask for a device with the given Virtual IP Address.

Table 56: Glossary

Term	Definition
VMAC Address	A Virtual MAC address is an address that is assigned by software to override the actual MAC address.
WebUI Port	This is the port on which the administration web server (WebUI) listens. For example, if you set this to 8090, you can connect to the DX by typing something like http://junipername.yourdomain.com:8090
WebUI SSL	Turn SSL on or off for the administration web server (WebUI). The first time, this must be performed in the CDXSHELL, and you will be prompted to generate a certificate.

Appendix B
List of Events

EMERG Events

- "DX Server was started"
- "Not licensed for this device"

Table 57: EMERG Events Messages

Message	Description
"ntp daemon was started"	The NTP process was started.
"admin server was started"	The WebUI was started
"ssh daemon was started"	The SSH server was started
"telnet daemon was started"	The telnet process was started.
"snmp daemon was started"	The SNMP process was started.
"DX Server was started"	DX appliance was started.
"Not licensed for this device"	The pac file is not licensed for this DX appliance.
"DX Server was started"	DX appliance was started.
"Warning: License key file failed"	Warning message to indicate that the license key file is missing.

ALERT Events

Table 58: ALERT Events Messages

Message	Description
"admin password changed"	The password for the Administrator was changed.
"Bad HTTP request: client sent an invalid header line: <http_header_line>"	An HTTP request with and invalid head was received.
"Bad HTTP request: HEAD/0.9"	HEAD request cannot be Version HTTP 0.9.
"Bad HTTP request: header line longer than allowed or poorly formed"	An HTTP request with a header line longer than allowed or a poorly formed HTTP request was received.
"Bad HTTP request: POST length is less than zero. Request line: <POST request_line>"	An HTTP request with the method POST that has a length less than zero was received.
"Bad HTTP request: POST request did not contain content length. Request line: <POST request_line"	An HTTP request with the method POST that did not contain the content length was received.
"Bad HTTP request: POST request specified content length of zero and is not configured to allow this"	An HTTP request with the method POST that specified the content length to be zero was received, but the DX appliance was not configured to allow zero length POST requests.

Table 58: ALERT Events Messages

Message	Description
“Bad or missing private key file <keypath>; password not set”	Invalid or missing private key file.
“Cannot contact Default Gateway <gateway>”	Cannot ping the gateway.
“Cannot contact DNS server <dns_server>”	Unable to contact the DNS server.
“Cannot contact E-mail server <email_server>”	Unable to contact the E-mail server.
“Cannot contact NTP server <ntp_server>”	Unable to contact the NTP server.
“Cannot contact syslog host <syslog_host>”	Unable to contact the syslog host.
“Cannot contact Target Server <target_server>”	Unable to contact the Target server.
“Cannot contact TFTP server <tftp_server>”	Unable to contact the TFTP server.
“Cannot upgrade: archive is <number_of_bytes> Kilo bytes. Flash has <number of bytes> available”	Insufficient space on the Flash to perform the upgrade.
“Cluster not in operation; there is no VIP present”	The cluster is missing the Virtual IP address.
“Duplicate entry found in the CRL file <crl_file>”	Duplicate entries were found in the CRL file.
“DX received excessive bytes from a target <target_server> for request <url_requested>”	The DX appliance received more bytes from a target server than is indicated in the HTTP header.
“Failed to add CA cert to trusted list: <internal error message>”	Unable to add the CA Certificate to the CA Trusted List.
“Failed to load cacrlfile <ca-crl_file>; check file format”	Unable to load the CA CRL file. The CA CRL file must be in a base64-encoded format.
“Failed to add CRL from cacrlfile <ca_crl_file>”	Unable to add the CRL to the CA CRL file.
“Failed to load the complete config”	Failed to load the configuration.
“Illegal Content-Length header of <length> sent from <target_server> for a request <url_requested>	Invalid content length sent from the Target server.
“Illegal replay from <target_server> (HTTP <http version>) for a request <url_requested> (no Content-length/chunking/connection: Close)”	Target server is HTTP1 and does not specify “connection: close” or “content length” or does not chunk.
“Illegal reply from <target_server> (HTTP <http version>) for a request <url_requested> (no Content-length/keep-alive set)”	The HTTP 1.0 Target server wants to do “keep-alive” but not without setting the “content-length” header.
“<IP address> transitioning to active	The DX appliance has transitioned from a standby role to active role.

Table 58: ALERT Events Messages

Message	Description
"Layer 2 Link Down on Main Interface"	The link was down on the main network interface, ether0.
"No client authentication CA certfile specified"	Missing CA Certificate file. CA Certificate file specifies the list of acceptable CA Certificates that a client may connect with.
"No clusters are in operation due to <configuration> errors"	All clusters are disabled.
"Only <number> of clusters out of <number> in operation"	Not all clusters are enabled.
"Rebooted from CLI"	The DX appliance was rebooted; initiated from the CLI.
"Target server <target_server> disabled through configuration"	Target server was disabled through the CLI or Web User interface.
"Target server <target_server> has been contacted"	Successfully established a TCP connection the Target server.
"Target server <target_server> passed Layer 7 health check"	Target server passed the Layer 7 health check performed by the DX appliance.
"The administrator password has been changed by pressing the reset button"	The reset button was pressed and thus the default administrator password was reset.
"The CA Trust file <ca_trust_file> could not be loaded; check file format"	Unable to load the CA Trust file. The CA Trust file must be in a base64-encoded format.
"The CA Certificate file <ca_cert_file> failed to load; check file format.	Unable to load the CA Certificate file. The CA Certificate file must be in a base64-encoded format.
"Threshold for the maximum number of connections exceeded"	The DX appliance has reached the threshold configured for the maximum number of connections.
"Received excessive bytes from the target <target_server> for a request <url_requested>"	Target server sent more bytes than what are specified in the "content-length" header.
"Rebooted from the CLI"	Accelerator was rebooted from the CLI.
"VIP <vip> down"	The VIP is down because all Target servers are down.
"VIP <vip> up"	The VIP is up.

Appendix C

Layer 4 Switching and ActiveN

This chapter describes Layer 4 Switching and ActiveN for the DX Application Acceleration Platform, discussing the following topics:

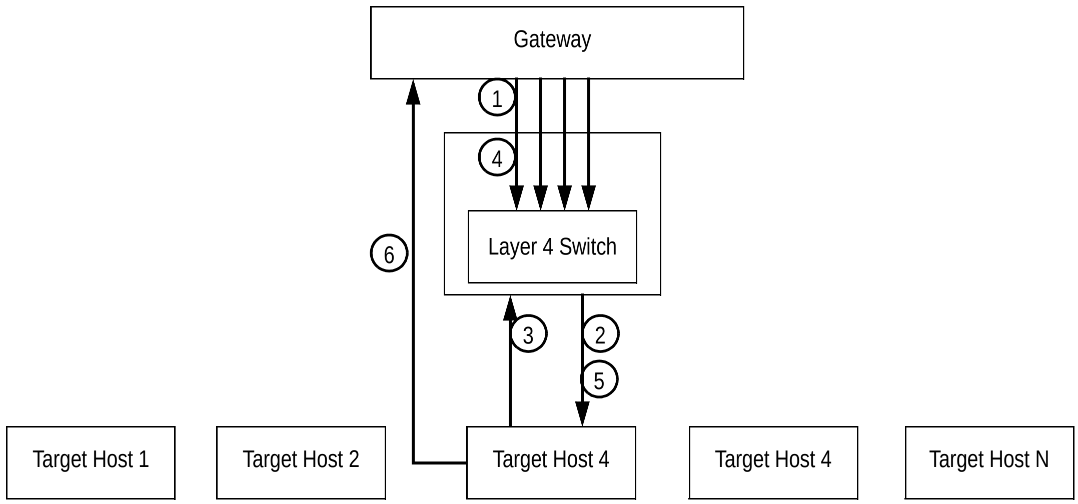
Overview

The ActiveN technology is based upon a Layer 4 switch that is build into each DX appliance. A Layer 4 Switch (L4S) is a packet-based switch based on the OSI "transport" layer. Layer 4 switches identify which application protocols (i.e., HTTP, SNTP, FTP, etc.) are included with each packet and uses this information to hand-off the packet to the appropriate blade or cluster.

The Layer 4 Switch Concept

Layer 4 switches are used to alleviate server loads by balancing traffic across a cluster of servers based upon individual session information and status. When an L4S is placed in front of cluster of servers running a particular application and a client makes a request for that application, the switch determines which server should handle the request, often based upon current server loads. Once the forwarding decision is made, the switch binds that session to a particular server. Figure 65 shows a typical model of Layer 4 switching with the target host configured for Direct Server Return (DSR).

Figure 65: Layer 4 Switching Example



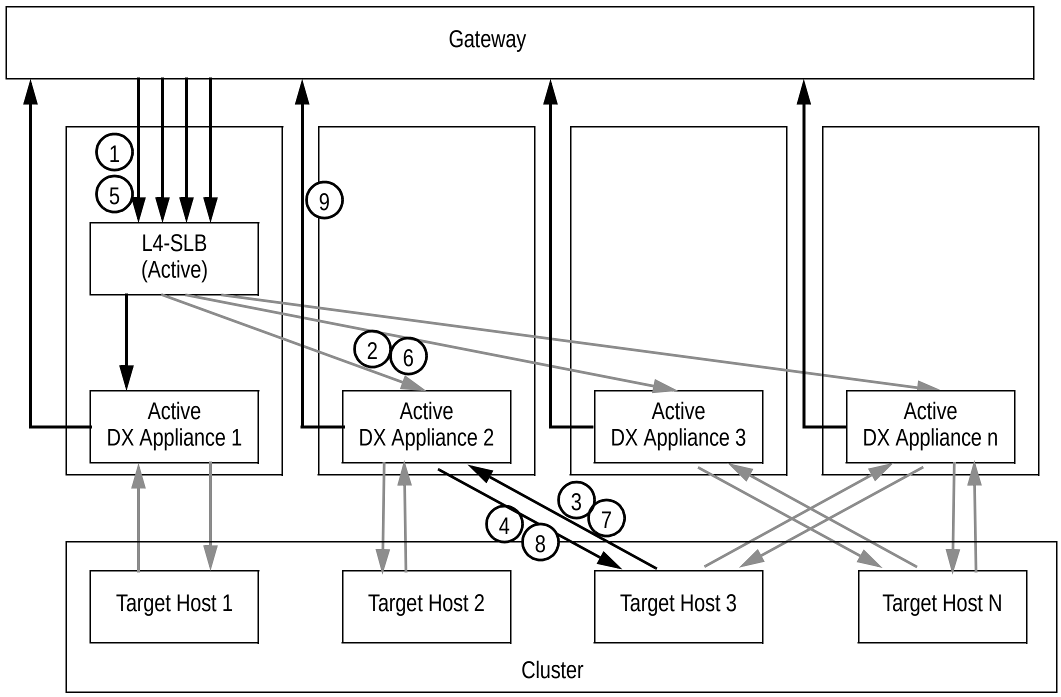
1. A request (SYN) arrives from the client.
2. The Layer 4 Switch forwards the request to the most available DX appliance (Target Host 3 in this example).
3. Target Host 3 terminates the connection and sends an acknowledgement (SYN-ACK) to the client.
4. The client sends a request.
5. The L4S forwards the request to Target Host 3.
6. Target Host 3 sends the response back directly to the client.

While this topology improves the performance of the site by implementing load balancing, it presents a single point of failure. If the L4S malfunctions for any reason, the site goes down.

Layer 4 Switching with Network Acceleration

Each Juniper DX appliance has a L4S built into it. This switch can be used in front of a group of DX appliances to act as a Server Load Balancer (SLB). The DX appliances are free to perform their normal acceleration operations. Figure 66 shows a topology where the L4S within the DX appliance is used for load balancing, and the target host configured for Direct Server Return (DSR).

Figure 66: Layer 4 Switching with Network Alteration Example



1. A request (SYN) arrives from the client.

2. The Layer 4 Switch (SLB) forwards the request to the most available DX appliance (the Juniper 2) for acceleration and distribution.

3. Juniper 2 forwards the request onto one of the target hosts within the cluster (Target Host 3 in this example).

4. Target Host 3 terminates the connection and sends an acknowledgement (SYN-ACK) to the client.

5. The client sends a request.

6. The L4S forwards the request to the same DX appliance (Juniper 2) for acceleration and distribution

7. Juniper 2 forwards the request onto one of the target hosts within the cluster (Target Host 3 in this example).

8. Target Host 3 sends the response back to the DX appliance (Juniper 2).

9. The DX appliance sends the response directly back to the client using Direct Server Return (DSR).

This topology improves the performance of the site by implementing load balancing and acceleration, but it still presents a single point of failure. If the L4S malfunctions for any reason, the entire site goes down. This is the problem that ActiveN technology was designed to prevent.

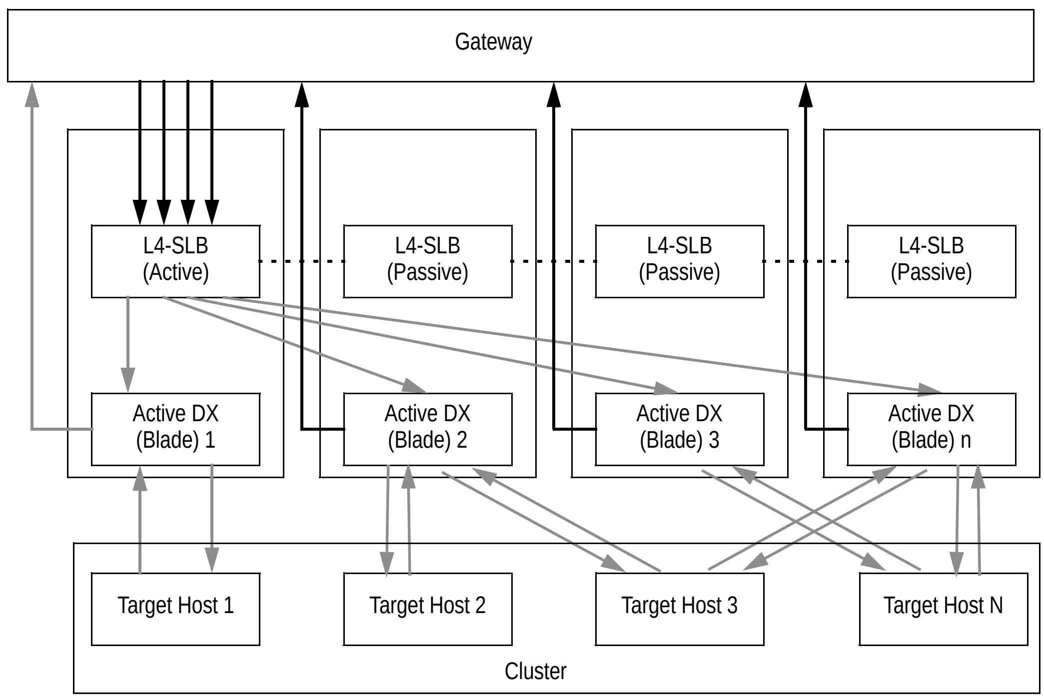
ActiveN Operation

ActiveN is designed to improve two aspects of networks operations:

- Reliability
- Scalability

The ActiveN topology uses the Layer 4 Switch (L4S) within the DX appliance to distribute user requests to configured DX appliances (also known as blades). An example of an ActiveN topology is shown in Figure 67.

Figure 67: Typical ActiveN Topology



ActiveN uses two different methods to improves network reliability and scalability; Failover at the L4S level, and Health Check at the Blade and Target Host levels.

- Failover is for L4S redundancy
- Health Check is to improve network reliability and scalability

Failover

In a healthy ActiveN configuration only one of the L4S performs the Server Load Balancing (SLB) operations (the active L4S). The L4S in each of the backup DX appliance (passive L4Ss) monitor the active L4S, and are ready to take over the L4S responsibilities immediately if a problem is detected. The L4S uses the same failover mechanism employed on the DX platform. The active L4S sends Redundancy Multicast Messaging Protocol (RMMP) health messages that the other L4S receives. If a certain number of health messages are not received within a time

window, the second L4S takes over the processing of new requests. (Note that the RMMP messages are actually passed at the Layer 2 level.)

The L4S uses a virtual MAC address. When the active L4S dies, the virtual MAC is removed from the interface and the backup L4S replaces it's real MAC address with the virtual one.

You can determine the failover state of a DX appliance by typing the command:

```
dx% show activeN failover
```

For example:

```
dx% show activeN failover
Failover: enabled
Mcast  addr: 239.0.0.1
Bind addr: not configured
Node Id: auto
Peer Port: 9199
Force master: disabled
Vmac: disabled
My node: 26890
Failover state: active
```

Layer 4 Switch Health Check

In order to properly balance traffic between the various DX appliance blades, the L4S must be aware of the health of each blade and remove them from rotation if they are not operating correctly. To monitor this, the DX appliance watches when a TCP connection is established to each DX appliance blade. If the connection is successful, the blade is operating. If the TCP connection fails, then the blade is considered down.

The L4S has a mechanism for finding the blades that belong to a group, determining their MAC address, and then determining their health. The user designates blades using the primary interface IP address/port for the particular blade (for example, "`172.16.0.10:80`" or "`172.16.0.10:443`"). This is the critical information that the L4S needs to determine the MAC address (for example, by using an ARP request to get the MAC). Once a MAC address is obtained for a blade and a successful TCP connection is established to the blade (as a health check), then the blade is officially rotated into the L4S group and it is ready to accept client requests.

You can determine the health state of a blade by typing the command:

```
dx% show activeN blade
______________________________
blade 1
Real IP: 10.0.201.18
Blade MAC: 0:e0:81:2e:c4:90
State: UP
______________________________
______________________________
blade 2
Real IP: 10.0.201.19
Blade MAC: 0:e0:81:2e:e2:3e
State: UP
______________________________
```

The line that says `State: UP indicates that the blade passed Layer 2 ARP learning`.

You can determine the health state of a group by typing the command:

```
dx% show activeN group
Group an_group
Vip: 10.0.201.20
Port: 443
Sticky: disabled
Total Blades: 2
Active Blades: 2
Blades:
Index  Status  Local   Real IP        Mac
1      UP      YES    10.0.201.18     00:e0:81:2e:c4:90
2      UP      NO     10.0.201.19     00:e0:81:2e:e2:3e
```

NOTE: Layer 4 Switch Failover and Layer 4 Switch Health Check are two separate and distinct processes. You can have a situation where the L4S are all reporting that they are enabled and working (active or standby), but the health check is down because the blades are non-responsive (either not working or not enabled).

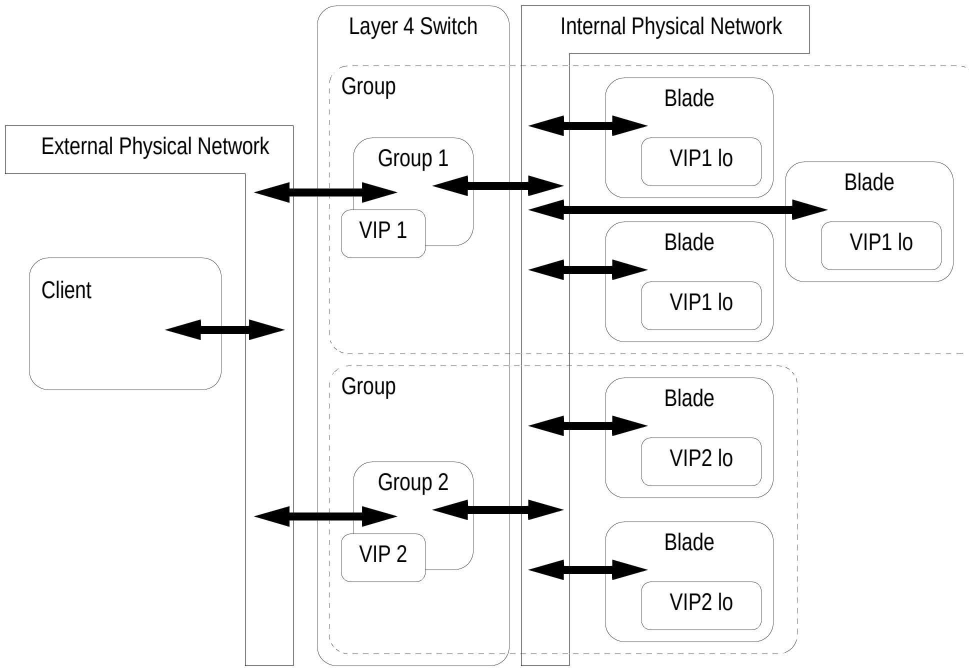
Port Symmetry

In order to minimize the amount of packet rewriting that the L4S must perform, the ActiveN VIP must match the Cluster VIP. This allows only the MAC to be rewritten (sometimes referred to as the MAC Address Translation, MAT), instead of requiring that the entire TCP layer be rewritten. This saves the checksum overhead incurred due to port rewriting. For example, if the L4S is advertising `192.168.10.100:80`, then the DX appliance blades in the corresponding group should be set to IP address `192.168.10.100` on loopback and listen on port 80.

Layer 4 Switch Grouping

Within the L4S, there is the concept of a "group" that is similar to the concept of a cluster that exists within the DX appliance. A group represents a collection of homogenous DX appliance blades, any one of which is capable of servicing a request. Load balancing rules are then applied to a particular group. Corresponding to each group is a virtual IP address (VIP) that is aliased on theL4S. Multiple groups can be created. Figure 68 shows this from a physical/logical combination.

Figure 68: Layer 4 Switch Groups

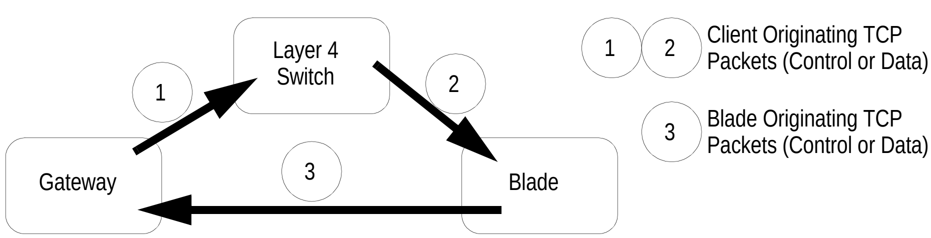


There is an internal physical network and an external physical network, where the internal network refers to the "backplane" of the DX appliance, and the external network is the customer network into which the DX appliance is being placed. In the example shown, the first group has three DX appliance blades while the second group has two. The VIPs for the DX appliance blades are placed on loopback (lo) while the VIPs for the groups in the L4S appear on the primary interface.

Connection Handling

The TCP connections made between the client, the L4S, and the target blades are asymmetric in nature. Client packets are sent to the L4S, but outbound target blade packets are sent directly back to the client. This implements a Direct Server Return (DSR) arrangement as shown in Figure 69.

Figure 69: DSR Operation



The L4S forwards packets from a client to the appropriate DX appliance blade. The packet forwarding mechanism operates on both new connections and existing ones. When a new connection comes in, as identified by a TCP SYN packet, the L4S must determine an appropriate destination. If there is no appropriate destination (as determined by all blades for a group being in a non-responsive state), then the packet is dropped. The packet is forwarded to the appropriate blade in the group. This can be programmed to be either the blade with the least number of outstanding connections, or each blade, in turn, in a round-robin fashion. The command for setting the switching policy is:

```
set activen advanced policy <leastconn | roundrobin>
```

Each connection is uniquely determined by its layer 3 and layer 4 components. The DX appliance uses a combination of the source IP/port and destination IP/port (although not together) to determine the appropriate destination. The first time a TCP connection comes in, the DX appliance uses the destination IP/port to look up first a group, and then a valid target blade MAC address. Subsequent packets (e.g., not TCP SYN packets) are mapped directly to the MAC address based on the source IP/port.

The L4S also monitors the packet flow for each TCP connection to determine when to purge the L4S client connection table entries. The difficulty in doing this lies in the fact that the L4S only sees half of the TCP session (the client's packets). In order to resolve a proper TCP teardown, the L4S must know whether the client initiated the close, or the server initiated the close. The DX appliance blades route their FIN and RST packets through the L4S. The L4S notes which TCP session the FIN or RST corresponds to, and forwards it on to the outbound gateway/router.

An aging system is also used to time-out entries in the L4S client connection table. This is because stale connections can expire due to lost hosts, etc. These stale connections accumulate over time and consume unnecessary resources.

Client IP Sticky

Client IP Sticky refers to a property of the load balancer where the same server is chosen for multiple TCP connections when the subsequent requests come from the same client. When a TCP connection arrives on a listen VIP:Port, the DX appliance looks in a “sticky entry table” to see if there is an entry for the client’s IP address. If there is an entry present in the table, then the server is retrieved and the session is created. If there is no entry is present in the sticky entry table, then the load balancing policy is applied and the selected server is listed in the sticky table along with the client’s IP address. A sticky entry is kept in the table until it exceeds the sticky timeout value set using the command:

```
set activeN group <name|all> blade sticky timeout <minutes>
```

The command `"set activeN sticky timeout"` is not per group, but rather it is a global command that affects all the groups.

There may be cases when a sticky entry could be deleted prematurely. One case is when the server goes down before the sticky timeout expires, and a new request from the same client arrives. In that case, the entry is flushed and a new server is fetched and re-inserted.

Index